

(12) 特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局



(43) 国際公開日
2011年5月5日(05.05.2011)

PCT

(10) 国際公開番号
WO 2011/052181 A1

- (51) 国際特許分類:
G06F 17/30 (2006.01)
- (21) 国際出願番号: PCT/JP2010/006305
- (22) 国際出願日: 2010年10月25日(25.10.2010)
- (25) 国際出願の言語: 日本語
- (26) 国際公開の言語: 日本語
- (30) 優先権データ:
特願 2009-246868 2009年10月27日(27.10.2009) JP
- (71) 出願人 (米国を除く全ての指定国について): 株式会社エスグランツ (S.GRANTS CO., LTD.) [JP/JP]; 〒2610004 千葉県千葉市美浜区高洲三丁目5番3棟1210号 Chiba (JP).
- (72) 発明者: および
- (75) 発明者/出願人 (米国についてのみ): 新庄 敏男 (SHINJO, Toshio) [JP/JP]; 〒2610004 千葉県千葉市美浜区高洲三丁目5番3棟1210号株式会社エスグランツ内 Chiba (JP). 國分 光裕 (KOKUBUN, Mitsuhiro) [JP/JP]; 〒2610004 千葉県千葉市美浜区高洲三丁目5番3棟1210号株式会社エスグランツ内 Chiba (JP).

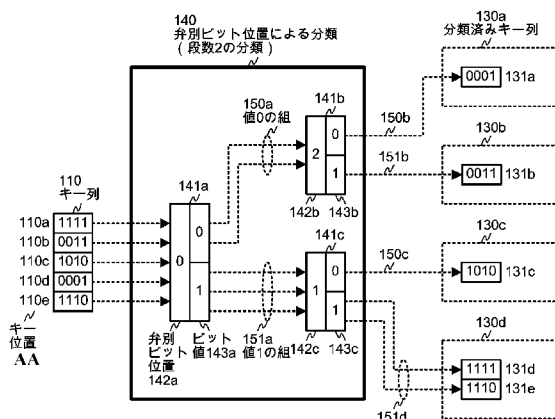
- (74) 代理人: ▲徳▼永 民雄(TOKUNAGA, Tamio); 〒2020012 東京都西東京市東町3丁目2番19号 Tokyo (JP).
- (81) 指定国 (表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.
- (84) 指定国 (表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), ヨーロッパ (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

[続葉有]

(54) Title: BIT-SEQUENCE KEY CLASSIFICATION/DISTRIBUTION DEVICE, CLASSIFICATION/DISTRIBUTION METHOD, AND PROGRAM

(54) 発明の名称: ビット列キー分類・分配装置、分類・分配方法及びプログラム

[図3]



110 Key array
130a Already-classified key array
140 Classification by discrimination bit position (two-level classification)
142a Discrimination bit position
143a Bit value
150a Keys with value 0
151a Keys with value 1
AA Key position

(57) Abstract: Provided is a classification technique that can classify keys into a plurality of blocks such that there is no overlap in key value ranges. Also provided is a technique for distributing keys classified by the provided classification technique. Said techniques use an adaptation of coupled-node tree technology. In the provided classification technique, classification keys are successively selected from a key memory means that stores keys to be classified, and the classification keys are used to generate a classification tree which is an adapted coupled-node tree. The keys are classified by associating each leaf node with keys, the keys being classified into N blocks. The number of levels in the classification tree is limited by the number (N) of blocks. Leaf nodes are extracted from the classification tree, and the associated keys are extracted and distributed as already-classified keys.

(57) 要約: キーを複数のブロックに分類するとき、キーの値の範囲に重複のない分類手法と分類されたキーを分配する手法を、カップルドノードツリーの技術を用いて提供する。分類対象のキーを記憶するキー記憶手段からキーを順次分類キーとして選択し、分類キーによりカップルドノードツリーを応用した分類ツリーを生成し、そのリーフノードをそれぞれN個のブロックに分類されるキーと対応付けることにより分類する。分類ツリーの段数は、ブロックの個数Nに応じて制限される。分類ツリーからリーフノードを取り出し、対応付けられたキーを分類済みキーとして取り出して分配する。



添付公開書類:

— 国際調査報告 (条約第 21 条(3))

明 細 書

発明の名称：

ビット列キー分類・分配装置、分類・分配方法及びプログラム

技術分野

[0001] 本発明は、分類対象のビット列キーを分類する技術と、分類されたキーを出力先に分配する技術に関する。

背景技術

[0002] 近年、社会の情報化が進展し、大規模なデータベースが各所で利用されるようになってきている。このような大規模なデータベースからレコードを検索するには、各レコードの記憶されたアドレスと対応づけられたレコード内の項目をインデックスキーとして検索をし、所望のレコードを探し出すことが通例である。また、全文検索における文字列も、文書のインデックスキーと見なすことができる。

[0003] そして、それらのインデックスキーはビット列で表現されることから、データベースの検索はビット列データの検索に帰着されるということができる。

一方、データベースに関連した処理として、下記特許文献 1 及び下記特許文献 2 に記載されているように、データベース中のレコードのマージソートが行われている。このマージソートもビット列データのマージソートに帰着される。

[0004] 基本的なマージソートの手法は、データを 2 個の対まで分割してそれを並び替え、並び替えられたものを組み合わせて行くものである。つまり、ソート対象データの分割を繰り返しながらソートして複数のソート済みのデータを得る前段の処理と、ソート済みのデータのマージを繰り返してソート対象データ全体のソートを行う後段の処理に分かれる。

特許文献 2 には、マージソートの後段の処理について図 1 に示す処理が開示されている。

[0005] 図1に示すように、ブロック1～ブロックNにはソート済みのデータが格納されており、ブロック1の最小値は13、ブロック2の最小値は8で、次に大きいデータは22であることが例示されている。同様に、ブロック3の最小値が53、ブロック4の最小値が24、ブロックNの最小値が9であると例示されている。

[0006] マージソートの後段の処理は、上記ブロック1～ブロックNの存在を前提としたものである。したがって、図1に示すマージソートを実行するためには、データをN個のブロックに分類することが必要である。

そして、図1に示すように、各ブロックから最小値を取り出し、各ブロックの最小値によるデータ列を生成することによりマージソートを実現している。

ところで、図1に示すように、従来のデータをN個のブロックに分類する手法では、各ブロックに分類されるデータの範囲に重複がある。したがって、各ブロックのサイズをキャッシュメモリに収まるものとすることにより、各ブロック内のソートを高速に実行できるとしても、後段の各ブロック内のデータをマージソートする処理では、ソート対象データ全体のソートを行うため、キャッシュメモリを用いた高速な処理とすることはできない。

[0007] 一方、ビット列データの検索に用いられるデータ構造として、下記特許文献3にカップルドノードツリーの配列に格納された例が開示されている。配列にカップルドノードツリーを格納するのは、ノードの位置を配列番号で表すことができ、代表ノードの位置を示す位置情報の情報量を削減することができるからである。

また、下記特許文献4には、カップルドノードツリーの任意の部分木の最小値や最大値を検索する手法や、カップルドノードツリーの任意の部分木から昇順あるいは降順でインデックスキーを取り出す手法が開示されている。

以下、図2A及び図2Bを参照して、本発明の背景技術の一つとして、カップルドノードツリーについて説明する。

[0008] 図2Aは、特許文献3に記載された配列に格納されたカップルドノードツ

リーの構成例を説明する図である。図2Aに示すように、ノード101が配列100の配列番号10の配列要素に配置されている。ノード101はノード種別102、弁別ビット位置103及び代表ノード番号104で構成されている。ノード種別102の値は0であり、ノード101がブランチノードであることを示している。弁別ビット位置103には1が格納されている。代表ノード番号104にはリンク先のノード対の代表ノードの配列番号20が格納されている。なお、以下では表記の簡略化のため、代表ノード番号に格納された配列番号を代表ノード番号ということもある。また、代表ノード番号に格納された配列番号をそのノードに付した符号あるいはノード対に付した符号で表すこともある。

[0009] 配列番号20の配列要素には、ノード対111の代表ノードであるノード[0]112が格納されている。そして隣接する次の配列要素（配列番号20+1）に代表ノードと対になるノード[1]113が格納されている。ノード[0]112はノード101と同様にブランチノードである。ノード[0]112のノード種別114には0が、弁別ビット位置115には3が、代表ノード番号116には30が格納されている。またノード[1]113は、ノード種別117と参照ポインタ118aで構成されている。ノード種別117には1が格納されており、ノード[1]113がリーフノードであることを示している。参照ポインタ118aには、インデックスキーの記憶領域を参照するポインタが格納されている。以下では表記の簡略化のため、参照ポインタに格納されたデータのことも参照ポインタという。

[0010] 配列番号30及び31の配列要素に格納されたノード122とノード123からなるノード対121の内容は省略されている。なお、代表ノードをノード[0]で表し、それと対になるノードをノード[1]で表すことがある。また、ある配列番号の配列要素に格納されたノードを、その配列番号のノードということがあり、ノードの格納された配列要素の配列番号を、ノードの配列番号ということもある。さらに、あるリーフノードと、そのリーフノードの参照ポインタが示す記憶領域に格納されたインデックスキーとの関係を

示すために、リーフノードに対応するインデックスキーということもあり、インデックスキーに対応するリーフノードということもある。

[0011] ノード [0] 1 1 2、ノード [1] 1 1 3、ノード 1 2 2、及びノード 1 2 3の格納された配列要素にそれぞれ付された0あるいは1は、検索キーで検索を行う場合にノード対のどちらのノードにリンクするかを示すものである。前段のブランチノードの弁別ビット位置にある検索キーのビット値である0か1を代表ノード番号に加えた配列番号のノードにリンクする。したがって、前段のブランチノードの代表ノード番号に、検索キーの弁別ビット位置のビット値を加えることにより、リンク先のノードが格納された配列要素の配列番号を求めることができる。

[0012] 図2Bは、カップルドノードツリーのツリー構造を概念的に示すとともにインデックスキーの記憶領域の例を示す図である。

符号4 1 0 aで示すのが図2Bに例示するカップルドノードツリー4 0 0のルートノードである。図示の例では、ルートノード4 1 0 aは配列番号4 2 0に配置されたノード対4 0 1 aの代表ノードとしている。ツリー構造としては、ルートノード4 1 0 aの下にノード対4 0 1 bが、その下層にノード対4 0 1 cとノード対4 0 1 fが配置され、ノード対4 0 1 fの下層にはノード対4 0 1 hとノード対4 0 1 gが配置されている。ノード対4 0 1 cの下にはノード対4 0 1 dが、さらにその下にはノード対4 0 1 eが配置されている。

[0013] 各ノードの前に付された0あるいは1の符号は、図2Aにおいて説明した配列要素の前に付された符号と同じである。検索キーの弁別ビット位置のビット値に応じてツリーをたどり、検索対象のインデックスキーに対応するリーフノードを見つけることになる。

[0014] 図示された例では、ルートノード4 1 0 aのノード種別4 6 0 aは0でブランチノードであることを示し、弁別ビット位置4 3 0 aは0を示している。代表ノード番号は4 2 0 aであり、それはノード対4 0 1 bの代表ノード4 1 0 bの格納された配列要素の配列番号である。

- [0015] ノード対4 0 1 bはノード4 1 0 bと4 1 1 bで構成され、それらのノード種別4 6 0 b、4 6 1 bはともに0であり、ブランチノードであることを示している。ノード4 1 0 bの弁別ビット位置4 3 0 bには1が格納され、リンク先の代表ノード番号にはノード対4 0 1 cの代表ノード4 1 0 cの格納された配列要素の配列番号4 2 0 bが格納されている。
- [0016] ノード4 1 0 cのノード種別4 6 0 cには1が格納されているので、このノードはリーフノードであり、したがって、参照ポインタ4 5 0 cを含んでいる。参照ポインタ4 5 0 cには、インデックスキー2 7 0 cが格納されている記憶領域を参照するポインタを格納する。参照ポインタ4 5 0 cに格納されたデータのこととも参照ポインタといい、符号4 8 0 cにより表す。他のリーフノードでも同様に、参照ポインタと参照ポインタに格納されたデータを同じ参照ポインタという語で表す。図2 Bに例示するインデックスキーの記憶領域3 1 1の参照ポインタ4 8 0 cが指す領域には、インデックスキー2 7 0 cとして“0 0 0 1 1 1”が記憶されている。
- [0017] ノード対4 0 1 cのもう一方のノード4 1 1 cのノード種別4 6 1 cは0、弁別ビット位置4 3 1 cは2であり、代表ノード番号にはノード対4 0 1 dの代表ノード4 1 0 dの格納された配列要素の配列番号4 2 1 cが格納されている。
- [0018] ノード4 1 0 dのノード種別4 6 0 dは0、弁別ビット位置4 3 0 dは5であり、代表ノード番号にはノード対4 0 1 eの代表ノード4 1 0 eの格納された配列要素の配列番号4 2 0 dが格納されている。ノード4 1 0 dと対をなすノード4 1 1 dのノード種別4 6 1 dは1であり、参照ポインタ4 5 1 dには、“0 1 1 0 1 0”というインデックスキー2 7 1 dを格納した記憶領域を示す参照ポインタ4 8 1 dが格納されている。
- [0019] ノード対4 0 1 eのノード4 1 0 e、4 1 1 eのノード種別4 6 0 e、4 6 1 eはともに1であり双方ともリーフノードであることを示している。ノード4 1 0 e、4 1 1 eの参照ポインタ4 5 0 e、4 5 1 eにはそれぞれ、値が“0 1 0 0 1 0”であるインデックスキー2 7 0 eと、“0 1 0 0 1 1

”であるインデックスキー271eを格納した記憶領域への参照ポインタ480e、481eが格納されている。

[0020] ノード対401bのもう一方のノードであるノード411bの弁別ビット位置431bには2が格納され、リンク先の代表ノード番号にはノード対401fの代表ノード410fの格納された配列要素の配列番号421bが格納されている。

ノード対401fのノード410f、411fのノード種別460f、461fはともに0であり双方ともブランチノードである。それぞれの弁別ビット位置430f、431fには5、3が格納されている。ノード410fの代表ノード番号にはノード対401gの代表ノード410gの格納された配列要素の配列番号420fが格納され、ノード411fの代表ノード番号にはノード対401hの代表ノードであるノード[0]410hの格納された配列要素の配列番号421fが格納されている。

[0021] ノード対401gのノード410g、411gのノード種別460g、461gはともに1であり双方ともリーフノードであることを示す。ノード410g、411gのそれぞれの参照ポインタ450g、451gには値が“100010”と“100011”であるインデックスキー270g、271gを格納した記憶領域への参照ポインタ480g、481gが格納されている。

[0022] また同じくノード対401hの代表ノードであるノード[0]410hとそれと対をなすノード[1]411hのノード種別460h、461hはともに1であり双方ともリーフノードであることを示している。ノード410h、411hのそれぞれの参照ポインタ450h、451hには値がそれぞれ“101011”と“101100”であるインデックスキー270h、271hを格納した記憶領域への参照ポインタ480h、481hが格納されている。

[0023] 上述のカップルドノードツリー400からインデックスキー“100010”を検索する処理の流れを簡単に説明する。弁別ビット位置は、左から0、

1、2、・・・とする。まず、ビット列“100010”を検索キーとしてルートノード410aから処理をスタートする。ルートノード410aの弁別ビット位置430aは0であるので、検索キー“100010”の弁別ビット位置が0のビット値をみると1である。そこで代表ノード番号の格納された配列番号420aに1を加えた配列番号の配列要素に格納されたノード411bにリンクする。ノード411bの弁別ビット位置431bには2が格納されているので、検索キー“100010”の弁別ビット位置が2のビット値をみると0であるから、代表ノード番号の格納された配列番号421bの配列要素に格納されたノード410fにリンクする。

[0024] ノード410fの弁別ビット位置430fには5が格納されているので、検索キー“100010”の弁別ビット位置が5のビット値をみると0であるから、代表ノード番号の格納された配列番号420fの配列要素に格納されたノード410gにリンクする。

[0025] ノード410gのノード種別460gは1でありリーフノードであることを示しているので、参照ポインタ480gにより示される記憶領域を参照し、そこに格納されたインデックスキー270gを読み出して検索キーと比較する。すると、インデックスキー270gと検索キーの両方とも値が“100010”であって一致している。このようにしてカップルドノードツリーを用いた検索が行われる。

[0026] 上述のカップルドノードツリー400を用いた検索は、8個のインデックスキーをそれぞれ8個のリーフノードに対応するグループに分類する処理と見ることができる。すなわち、特許文献3には、ブロック内のインデックスキーが1個であるインデックスキーの分類手法が開示されていると考えることもできる。

しかしながら、上述のマージソートの後段の処理の前提となる、ブロック内のインデックスキーが複数であるインデックスキーの分類手法は開示されていない。

先行技術文献

特許文献

[0027] 特許文献1：特開2000-010761号公報

特許文献2：特開2006-163565号公報

特許文献3：特開2008-269503号公報

特許文献4：特開2008-112240号公報

発明の概要

発明が解決しようとする課題

[0028] そこで、本発明が解決しようとする課題は、ビット列からなるデータ（以下、ビット列キー、あるいは単にキーということがある。また、インデックスキーということもある。）を複数のブロックに分類するとき、キーの値の範囲に重複のない分類手法と分類されたビット列キーを出力先に分配する手法を、カップルドノードツリーの技術を応用して提供することである。

課題を解決するための手段

[0029] 本発明によるN個のブロックへのビット列キーの分類手法は、分類対象のキーを記憶するキー記憶手段からキーを順次分類キーとして選択し、分類キーによりカップルドノードツリーを応用した分類ツリーを生成し、そのリーフノードをそれぞれN個のブロックに分類されるキーと対応付けることにより行う。分類ツリーの段数は、ブロックの個数Nに応じて制限される。

本発明の一つの態様によれば、分類ツリーのリーフノードは、キー記憶手段に記憶されている分類キーの位置情報を取得するために用いるキーアクセス情報を含む。そして、キーアクセス情報を用いてキーの位置情報を取得するためのキー位置検索表を生成する。

分類ツリーの段数 n は、2の $(n-1)$ 乗がブロックの個数 N と等しいかそれより大きい最小の値に制限される。例えば $N=8$ であれば、 $n=4$ となる。

[0030] 本発明の一つの態様によれば、分類対象のキーすべてにより生成された分類ツリーから、順次リーフノードを取り出し、リーフノードから読み出したキ

ーアクセス情報を用いてキー位置検索表よりリーフノードに対応するブロックに分類されたキーの位置情報を得て、キー記憶手段からキーを読み出して出力先に出力する。

発明の効果

- [0031] カップルドノードツリーの特徴により、あるリーフノードの直近上位のブランチノードの弁別ビット位置までの上位のビット値は、同一のブロックに分類されるどのキーでも一致し、リーフノードの直近上位のブランチノードの弁別ビット位置までに異なるビット値のあるキーは、他のブロックに分類される。したがって、本発明によれば、カップルドノードツリーの構造を持つ分類ツリーを生成することにより、分類対象のキーの値の範囲に重複がないように、キーを効率的に分類することができる。

図面の簡単な説明

- [0032] [図1] ソート済みのデータのマージを繰り返してソート対象データ全体のソートを行う処理を説明する図である。
- [図2A] 配列に格納されたカップルドノードツリーの構成例を説明する図である。
- [図2B] カップルドノードツリーのツリー構造を概念的に示す図である。
- [図3] 本発明の一実施の形態における分類処理の概念を説明する図である。
- [図4] 本発明を実施するためのハードウェア構成例を説明する図である。
- [図5] 本発明の一実施の形態に係る分類ツリーのツリー構造を概念的に示すとともに、インデックスキーの管理領域を説明する図である。
- [図6A] 分類キー“010011”を挿入する前の状態を説明する図である。
- [図6B] 分類キー“010011”の挿入後であって分類キー“010010”を挿入する前の状態を説明する図である。
- [図6C] 分類キー“010010”の挿入後であって分類キー“000111”を挿入する前の状態を説明する図である。
- [図6D] ノード対を挿入するため挿入位置のリーフノードのキーリンク表にインデックスキーを連結した状態を説明する図である。

[図6E] 挿入位置のリーフノードのキーリンク表にインデックスキーを連結した後であって分類キー“000111”を挿入した状態を説明する図である。

[図7A] 本発明の一実施の形態における分類処理の処理フロー例を示す図である。

[図7B] 本発明の一実施の形態における分配処理の処理フロー例を示す図である。

[図8] 本発明の一実施の形態における分類キーにより分類ツリーを生成する処理フロー例を説明する図である。

[図9A] 本発明の一実施の形態における分類キーを分類ツリーに挿入する処理の初段の処理フロー例を示す図である。

[図9B] 本発明の一実施の形態における分類キーを分類ツリーに挿入する処理の中段の処理フロー例を示す図である。

[図9C] 本発明の一実施の形態における分類キーを分類ツリーに挿入する処理の後段の処理フロー例を示す図である。

[図10] 本発明の一実施の形態における分類キーを挿入位置に挿入する処理フロー例を示す図である。

[図11] 本発明の一実施の形態における分類キーを、リーフノードのキーリンク表に連結する処理フロー例を示す図である。

[図12] 本発明の一実施の形態における分類ツリーを探索してリーフノードのキーリンク表に分類キーを連結する処理フロー例を説明する図である。

[図13] 本発明の一実施の形態におけるインデックスキーの最小値を含むノードを検索する処理フロー例を説明する図である。

[図14] 本発明の一実施の形態におけるリーフノードのノード対の分類参照ポインタを連結する処理フロー例を説明する図である。

[図15A] 本発明の一実施の形態における、インデックスキーの最小値を含むリーフノードを求め、リーフノードのキーリンク表に連結したインデックスキーを順次取り出す処理の前段の処理フロー例を説明する図である。

[図15B]本発明の一実施の形態における、インデックスキーの最小値を含むリーフノードを求め、リーフノードのキーリンク表に連結したインデックスキーを順次取り出す処理の後段の処理フロー例を説明する図である。

[図16]本発明の一実施の形態におけるビット列キー分類・分配装置の機能ブロック構成例を説明する図である。

発明を実施するための形態

[0033] 図3は、本発明の一実施の形態における分類処理の概念を説明する図である。図3に例示するものは、4つのブロックにキー进行分类するものである。したがって、分類ツリーは2段である。

[0034] 図3には、分類対象であるキーからなるキー列110が示されている。図3の例示では、キー列110に含まれるキーの存在するキーの位置であるキー位置が110a（以下、キー位置110aのように表記する。）である記憶領域にはキー“1111”が存在する。また、キー位置110b、110c、110d、110eには、それぞれキー“0011”、“1010”、“0001”、“1110”が存在する。

[0035] 図3に示す例では、キー列110に含まれるキー进行分类済みキー列130a、130b、130c、130dの4つのブロックに分类するため、弁別ビット位置による分類（段数2の分類）140により、分类される。仮に3つのブロックに分类する場合でも、弁別ビット位置による分類（段数2の分類）140により4つのブロックに分类してから隣接する2つのブロックに分类されたものを1つのブロックに分类されたものとして扱えばよいから、ブロック数を2のn乗のものとしても一般性を失わない。

[0036] 弁別ビット位置による分類（段数2の分類）140の第1段の弁別ビット位置による分類141aで分類に用いる弁別ビット位置142aは0である。インデックスキーのうち、ビット位置0のビット値143aの値が0のものは、点線の矢印の組150aが示すように、第2段の弁別ビット位置による分類141bでさらに分类される。

一方、インデックスキーのうち、ビット位置0のビット値143aの値が1のものは、点線の矢印の組151aが示すように、第2段の弁別ビット位置による分類141cでさらに分類される。

[0037] 第2段の弁別ビット位置による分類141bで分類に用いる弁別ビット位置142bは2である。第2段の弁別ビット位置による分類141bの分類対象である分類キーのうち、ビット位置2のビット値143bの値が0のインデックスキー“0001”は、点線の矢印150bが示すように、分類済みキー列130aに分類済みキー131aとして格納される。また、第2段の弁別ビット位置による分類141bの分類対象であるインデックスキーのうち、ビット位置2のビット値143bの値が1のインデックスキー“0011”は、点線の矢印151bが示すように、分類済みキー列130bに分類済みキー131bとして格納される。

[0038] 一方、第2段の弁別ビット位置による分類141cで分類に用いる弁別ビット位置142cは1である。第2段の弁別ビット位置による分類141cの分類対象であるインデックスキーのうち、ビット位置1のビット値143cの値が0のインデックスキー“1010”は、点線の矢印150cが示すように、分類済みキー列130cに分類済みキー131cとして格納される。また、第2段の弁別ビット位置による分類141cの分類対象であるインデックスキーのうち、ビット位置1のビット値143cの値が1の分類キー“1111”と“1110”は、点線の矢印の組151dが示すように、分類済みキー列130dに、それぞれ分類済みキー131d、131eとして格納される。

[0039] 図4は、本実施形態のためのハードウェア構成例を説明する図である。

図4には、ビット列キー分類・分配装置300がネットワーク347を介してビット列キーソート装置340a、340b、・・・、340mに接続されたシステムが例として記載されている。ビット列キー分類・分配装置300による分類済みのビット列キーの分配先がビット列キーソート装置であるのは1つの例であり、集計処理等の他のアプリケーションにも適用可能であ

る。

また、分配先の装置も、ネットワークを介して接続されたものに限らず、例えばマルチプロセッサシステムの中央処理装置とすることもできる。

[0040] 本実施形態によるキーの分類は中央処理装置 302 及びキャッシュメモリ 303 を少なくとも備えたデータ処理装置 301 によりデータ格納装置 308 を用いて実施される。また、例示されたビット列ソート装置 340a、340b、・・・、340m への分類済みキーの分配も、データ処理装置 301 によりデータ格納装置 308 を用いて実施される。分類ツリーが記憶される配列 309、分類ツリーを検索中にたどるノードが配置された配列要素の配列番号を格納する探索経路スタック 310、分類対象であるキーが記憶されているインデックスキーの記憶領域 311、及び各ブロックに分類されるキーの位置情報を検索するデータを記憶するインデックスキーの管理領域 320 を有するデータ格納装置 308 は、主記憶装置 305 または外部記憶装置 306 で実現することができる。

[0041] ビット列ソート装置 340a、340b、・・・、340m はそれぞれ分配された分類済みキーを並列にソートする。これらの構成例として、ビット列ソート装置 340a の構成例が示されている。図 4 に示すように、ビット列ソートは、中央処理装置 302 及びキャッシュメモリ 303 を少なくとも備えたデータ処理装置 301a によりデータ格納装置 308a を用いて実施される。配列 309a、インデックスキーの記憶領域 310a を有するデータ格納装置 308a は、主記憶装置 305a または外部記憶装置 306a で実現することができる。

[0042] 図 4 の例示では、ビット列分類・分配装置 300 の主記憶装置 305、外部記憶装置 306 及び通信装置 307 が一本のバス 304 によりデータ処理装置 301 に接続されているが、接続方法はこれに限るものではない。また、主記憶装置 305 をデータ処理装置 301 内のものとすることもできるし、探索経路スタック 310 を中央処理装置 302 内のハードウェアとして実現することも可能である。あるいは、インデックスキーの記憶領域 311 は外

部記憶装置 306 に、探索経路スタック 310 とインデックスキーの管理領域 320 を主記憶装置 305 に持つなど、使用可能なハードウェア環境、インデックスキー集合の大きさ等に応じて適宜ハードウェア構成を選択できることは明らかである。

[0043] なお、特に図示されてはいないが、処理の途中で得られた各種の値や初期設定値等を後の処理で用いるためにそれぞれの処理に応じた主記憶装置 305 の一時記憶領域が用いられることは当然である。

また、ビット列ソート装置 340a、340b、・・・、340m における各装置の接続方法やデータ格納装置の構成についても同様である。

[0044] 図 5 は、本発明の一実施の形態に係る分類ツリーのツリー構造を概念的に示すとともに、インデックスキーの管理領域を説明する図である。

図 5 に例示する分類ツリー 200 は、図 2B に例示するインデックスキーの記憶領域 311 に記憶されたインデックスキーを 4 つのブロックに分類するためのものである。

図 5 に示すインデックスキーの管理領域 320 には、キー分類表 321 とキーリンク表 322 が格納されている。キー分類表 321 とキーリンク表で、先に述べたキー位置検索表の例を構成しており、キー分類表 321 のエントリを指す分類参照ポインタは、先に述べたキーアクセス情報の例である。

図 5 のインデックスキーの管理領域 320 に記載された 270h 等の符号は、図 2B のインデックスキーの記憶領域 311 に記載された 270h 等の符号で示すインデックスキー “101011” 等を示している。

[0045] 符号 210a で示すのが図 5 に例示する分類ツリー 200 のルートノードである。図示の例では、ルートノード 210a は配列番号 220 に配置されたノード対 201a の代表ノードとしている。ツリー構造としては、ルートノード 210a の下にノード対 201b が、その下層にノード対 201c とノード対 201f が配置されている。

[0046] 図示された例では、ルートノード 210a のノード種別 260a は 0 でブランチノードであることを示し、弁別ビット位置 230a は 0 を示している。

代表ノード番号は220aであり、それはノード対201bの代表ノード210bの格納された配列要素の配列番号である。

[0047] ノード対201bはノード210bと211bで構成され、それらのノード種別260b、261bはともに0であり、ブランチノードであることを示している。ノード210bの弁別ビット位置230bには1が格納され、リンク先の代表ノード番号にはノード対201cの代表ノード210cの格納された配列要素の配列番号220bが格納されている。

[0048] ノード210cのノード種別260cには1が格納されているので、このノードはリーフノードである。分類ツリーのリーフノードは、分類参照ポインタ250cを含んでいる。分類参照ポインタ250cには、インデックスキーの管理領域320に含まれるキー分類表のエントリを指すポインタが格納される。分類参照ポインタ250cに格納されたデータも分類参照ポインタといい、符号280cにより表す。他のリーフノードでも同様に、分類参照ポインタと分類参照ポインタに格納されたデータを同じ分類参照ポインタという語で表す。

[0049] ノード対201cのもう一方のノードであるノード211cのノード種別261cにも1が格納されているので、このノードもリーフノードである。ノード211cの分類参照ポインタ251cには、分類参照ポインタ280eが格納されている。

[0050] ノード対201bのもう一方のノードであるノード211bの弁別ビット位置231bには2が格納され、リンク先の代表ノード番号にはノード対201fの代表ノードであるノード[0]210fの配置された配列要素の配列番号221bが格納されている。

ノード対201fの代表ノードであるノード[0]210fとそれと対をなすノード[1]211fのノード種別260f、261fはともに1であり双方ともリーフノードである。ノード210f、211fの分類参照ポインタ250f、251fには、それぞれ分類参照ポインタ280g、280hが格納されている。

- [0051] 図5に示すように、分類キーを4つのブロックに分類するための分類ツリー200は、第1段のブランチノードであるルートノード210aと第2段のブランチノードであるノード210bとノード211b、及び第3段目のノードとして、各ブロックに対応する4つのリーフノード210c、211c、210f、211fを含む。以下の説明において、リーフノードに対応するブロックに分類されるインデックスキーを、リーフノードに含まれるインデックスキーということがある。
- [0052] 一方、インデックスキーの管理領域320は、キー分類表321とキーリンク表322を含む。図5に示すインデックスキーの管理領域320の各データの記憶状況は、図2Bに示すインデックスキーの記憶領域311に記憶された全てのインデックスキーを4つのブロックに分類したときのものである。
- [0053] キー分類表321は、インデックスキーを分類するブロック数に対応して4つのエントリを有する。各エントリの先頭アドレスは、分類ツリー200の4つのリーフノード210c、211c、210f、211fの分類参照ポインタ280c、280e、280g、280hでそれぞれ示される。
- [0054] また、キー分類表321は、図の例では、各エントリに最小値キー312a、最大値キー312b、キーの出力先312e、先頭リンク312c、及び末尾リンク312dを含む。図に示す例では、分類参照ポインタ280cの指すエントリの先頭リンク312cと末尾リンク312dには、どちらにもキーリンク表322のエントリを指すキー管理ポインタ370cが格納されている。分類参照ポインタ280eの指すエントリの先頭リンク312cと末尾リンク312dには、それぞれキー管理ポインタ370eとキー管理ポインタ371dが格納されている。分類参照ポインタ280gの指すエントリの先頭リンク312cと末尾リンク312dには、それぞれキー管理ポインタ370gとキー管理ポインタ371gが格納されている。分類参照ポインタ280hの指すエントリの先頭リンク312cと末尾リンク312dには、それぞれキー管理ポインタ370hとキー管理ポインタ371hが格納

されている。

最小値キー 3 1 2 a、最大値キー 3 1 2 b、キーの出力先 3 1 2 e に格納される値の記載については省略されている。最小値キー 3 1 2 a と最大値キー 3 1 2 b の値は分類ツリー 2 0 0 が生成される途中で書き込まれ、更新されるが、キーの出力先 3 1 2 e は、分類ツリーの生成が終了した後に書き込まれる。なお、キーの出力先については、キー分類表とは別に、分類対象のキーを分類するブロック対応に設定しておくこともできる。

[0055] キーリンク表 3 2 2 は、同一のリーフノードに対応付けられるインデックスキーを順次たどることを可能とするためのインデックスキー間のリンク関係を書き込むものであり、分類対象のインデックスキーの数に対応したエントリを有する。例えば、インデックスキーの記憶領域が図 2 B に示すものであれば、8 つのエントリを有する。各エントリの先頭アドレスは、キー管理ポインタ 3 7 0 c、3 7 0 e、3 7 1 e、3 7 1 d、3 7 0 g、3 7 1 g、3 7 0 h、3 7 1 h で示されており、それぞれインデックスキー 2 7 0 c、2 7 0 e、2 7 1 e、2 7 1 d、2 7 0 g、2 7 1 g、2 7 0 h、2 7 1 h に対応付けられている。

[0056] キーリンク表 3 2 2 は、図の例では、各エントリにキー参照ポインタ 3 1 3 a とリンク 3 1 3 b を含む。キー参照ポインタ 3 1 3 a は、そのエントリを指すキー管理ポインタに対応付けられたインデックスキーの記憶領域を指す。したがって、インデックスの記憶領域が図 2 B に示すものであれば、キー管理ポインタ 3 7 0 c、3 7 0 e、3 7 1 e、3 7 1 d、3 7 0 g、3 7 1 g、3 7 0 h、3 7 1 h の指すキーリンク表 3 2 2 のキー参照ポインタ 3 1 3 a には、参照ポインタ 4 8 0 c、4 8 0 e、4 8 1 e、4 8 1 d、4 8 0 g、4 8 1 g、4 8 0 h、4 8 1 h が格納されるが、図 5 ではキー参照ポインタ 3 1 3 a の記載は省略されている。

[0057] キー管理ポインタの指すキーリンク表 3 2 2 のリンク 3 1 3 b には、キー管理ポインタに対応付けられたインデックスキーと同一ブロックに分類されるインデックスキーが対応付けられたキー管理ポインタが格納されている。図

5の例示では、キー管理ポイント 370 h に対応付けられたインデックスキー 270 h とキー管理ポイント 371 h に対応付けられたインデックスキー 271 h は、分類参照ポイント 280 h に対応するブロックに分類されている。そのブロックの先頭リンク 312 c であるキー管理ポイント 370 h の指すキーリンク表 322 のリンク 313 b には、キー管理ポイント 371 h が格納されている。インデックスキー 271 h に対応付けられたキー管理ポイント 371 h はキー分類表 321 の末尾リンクに格納され、インデックスキー 271 h はブロックの末尾のインデックスキーであるので、キー管理ポイント 371 h の指すキーリンク表 322 のリンク 313 b にはなにも格納されていない。

[0058] 同様に、キー管理ポイント 370 g に対応付けられたインデックスキー 270 g とキー管理ポイント 371 g に対応付けられたインデックスキー 271 g は、分類参照ポイント 280 g に対応するブロックに分類されている。そのブロックの先頭リンク 312 c であるインデックスキー 270 g に対応付けられたキー管理ポイント 370 g の指すキーリンク表 322 のリンク 313 b には、キー管理ポイント 371 g が格納されている。インデックスキー 271 g に対応付けられたキー管理ポイント 371 g はキー分類表 321 の末尾リンクに格納され、インデックスキー 271 g はブロックの末尾のインデックスキーであるので、キー管理ポイント 371 g の指すリンク 313 b にはなにも格納されていない。

[0059] キー管理ポイント 370 e に対応付けられたインデックスキー 270 e とキー管理ポイント 371 e に対応付けられたインデックスキー 271 e、及びキー管理ポイント 371 d に対応付けられたインデックスキー 271 d は、分類参照ポイント 280 e に対応するブロックに分類されている。そのブロックの先頭リンク 312 c であるインデックスキー 270 e に対応付けられたキー管理ポイント 370 e の指すキーリンク表 322 のリンク 313 b には、キー管理ポイント 371 e が格納されている。キー管理ポイント 371 e の指すキーリンク表 322 のリンク 313 b には、キー管理ポイント 37

1 d が格納されている。そして、インデックスキー 2 7 1 d に対応付けられたキー管理ポインタ 3 7 1 d はキー分類表 3 2 1 の末尾リンクに格納され、インデックスキー 2 7 1 d はブロックの末尾のインデックスキーであるので、キー管理ポインタ 3 7 1 d の指すキーリンク表 3 2 2 のリンク 3 1 3 b にはなにも格納されていない。

[0060] また、キー管理ポインタ 3 7 0 c に対応付けられたインデックスキー 2 7 0 c は、分類参照ポインタ 2 8 0 c に対応するブロックに分類されている。分類参照ポインタ 2 8 0 c の指すキー分類表 3 2 1 の先頭リンク 3 1 2 c と末尾リンク 3 1 2 d にはともにインデックスキー 2 7 0 c に対応付けられたキー管理ポインタ 3 7 0 c が格納されているので、分類参照ポインタ 2 8 0 c に対応するブロックに分類されているインデックスキーはインデックスキー 2 7 0 c だけである。したがって、キー管理ポインタ 3 7 0 c が指すキーリンク表 3 2 2 のリンク 3 1 3 b には何も格納されていない。

なお、以下の説明において、リーフノードにインデックスキーを対応付けることを、インデックスキーをリーフノードのキーリンク表に連結するということがある。また、インデックスキーが対応付けられたリーフノードを、そのインデックスキーと連結するリーフノードということがある。

[0061] 上述のキー分類表 3 2 1 の分類参照ポインタに対応するブロックのインデックスキーは、次のようにして全て取り出すことができる。まず、分類参照ポインタの指す先頭リンク 3 1 2 c をキーリンク表 3 2 2 のキー管理ポインタとしてキー参照ポインタ 3 1 3 a を読み出し、キー参照ポインタ 3 1 3 a の指すインデックスキーをインデックスキー記憶領域から取り出す。次にリンク 3 1 3 b をキーリンク表 3 2 2 のキー管理ポインタとしてキー参照ポインタ 3 1 3 a を読み出し、キー参照ポインタ 3 1 3 a の指すインデックスキーをインデックスキー記憶領域から取り出すことを、リンク 3 1 3 b がキー分類表 3 2 1 の末尾リンク 3 1 2 d と等しくなるまで繰り返す。

[0062] 次に、図 6 A ～図 6 E を参照して、分類ツリー 2 0 0 の生成処理の概要を説明する。分類ツリー 2 0 0 のうちノード 2 1 1 b 以下の階層の部分につい

ての説明は省略する。また、すでにインデックスキー 271d “011010” がリーフノード 210b に対応するブロックに分類されている、すなわちリーフノード 210b のキーリンク表に連結されているものとする。なお、以下の説明においては、インデックスキーの記憶領域から読み出され、分類処理の対象となるインデックスキーを、分類キーということがある。また、分類キーを分類ツリーによりリーフノードに対応するブロックに分類すること、すなわちリーフノードのキーリンク表に連結することを、分類ツリーに分類キーを挿入するということがある。分類ツリーの最大段数に制限のあることから、分類ツリーに分類キーを挿入するとき、分類キーと連結するリーフノードを含むノード対を分類ツリーに挿入する場合のほか、分類キーを既存のリーフノードに連結する場合と、既存のリーフノードからなるノード対の直近上位のノードである親ノードをリーフノードとし、そのリーフノードのキーリンク表に分類キーを連結する場合がある。

[0063] 図 6A は、分類キー 270e “010011” を挿入する前の状態である分類ツリー 200a を説明する図である。図 6A に示す分類ツリー 200a は、図 5 に示す分類ツリー 200 のノード 210b がリーフノードとなったものであり、その分類参照ポインタ 250b には、分類参照ポインタ 280d が格納されている。また、ノード 211b より下位のノードの記載は省略されている。この下位ノードの記載の省略は、図 6B～図 6E においても同様である。

[0064] 図 6A の点線の矢印 290d で示すように、リーフノード 210b には分類参照ポインタ 280d によりインデックスキーの管理領域 320 を介してインデックスキー記憶領域 311 に記憶されたインデックスキー 271d “011010” が対応付けられている。言い換えれば、リーフノード 210b のキーリンク表にインデックスキー 271d が連結されている。

[0065] 図 6A には、挿入される分類キーとして分類キー 271e “010011” とその挿入位置が記載されている。

また、探索経路カウンタのカウント値が記載されている。探索経路カウンタ

のカウント値は、分類ツリーに分類キーを挿入するときに、分類ツリーの段数の制限を満足しながら、分類キーと連結するリーフノードを含むノード対を分類ツリーに挿入することができるかの判断に用いられる。

探索経路カウンタのカウント値は、例えば初期値を0としてカウントアップし、図6Aに示すように、ルートノードにおいて値1を取るものとする。なお、初期値を分類ツリーの最大段数としてカウントダウンし、カウント値が0か判定をすることにより、同じく、分類ツリーの段数の制限を満足しながら、分類キーと連結するリーフノードを含むノード対を分類ツリーに挿入することができるかの判断を行えることは明らかである。

[0066] 図6Aの例では、ルートノード210aの弁別ビット位置260aに格納された値が0、挿入される分類キー271e “010011”の0ビット目のビット値が0なので、挿入される分類キー271eによる検索の結果、リーフノード210bが求められる。探索経路カウンタの値はリーフノード210bにおいて2となる。また、分類キー271e “010011”とリーフノード210bのキーリンク表に連結したインデックスキー271d “011010”の上位ビット位置から見て最初に異なるビット値となるビット位置である差分ビット位置と、ブランチノード210aの弁別ビット位置の相対的位置関係により、分類キー271eはリーフノード210bの下位に挿入される。したがって、分類キー271eの挿入位置は、図に示すようにノード210bとなる。分類ツリーの最大段数は3、探索経路カウンタのカウント値は2であるので、段数の制限を満足することから、挿入される分類キー271eと連結するリーフノードを含むノード対を、ノード210bの下位に挿入することが可能である。

[0067] 図6Bは、分類キー271e “010011”の挿入後であって分類キー270e “010010”を挿入する前の状態である分類ツリー200bを説明する図である。分類キー271e “010011”と連結するリーフノードを含むノード対201dはノード210bの下位に挿入されている。分類ツリー200aではリーフノードであったノード210bはブランチノード

ドとなっており、弁別ビット位置には、インデックスキー 271d と分類キー 271e の差分ビット位置 2 が格納されている。ノード 210b の代表ノード番号 220b は挿入されたノード対 201d の代表ノード 210d が配置された配列要素の配列番号である。

[0068] インデックスキー 271d が分類キー 271e より大きいので、分類キー 271d と連結するリーフノードはノード対 201d のノード [1] であるノード 211d となる。図 6B に示すように、図 6A に示すノード 210b の内容が、図 6B のノード 211d にコピーされている。

[0069] 一方、ノード対 201d のノード [0] であるノード 210d が、分類キー 271e と連結するリーフノードとなる。その分類参照ポインタ 250d には、分類参照ポインタ 280e が格納されている。そして、点線の矢印 290e で示すように、リーフノード 210d には分類参照ポインタ 280e によりインデックスキーの管理領域 320 を介してインデックスキー記憶領域 311 に記憶された分類キー 271e “010011” が対応付けられている。すなわち、分類キー 271e “010011” は、リーフノード 210d に分類される。

[0070] 図 6B には、分類ツリー 200b に加えて、次に挿入される分類キーとして、分類キー 270e “010010” が記載され、分類キー 270e の挿入位置としてノード 210d が示されている。また、探索経路カウンタのカウント値はリーフノード 210d において最大段数である 3 と表示されている。したがって、段数制限により、ノード 210d の下位に分類キー 270e と連結するリーフノードを含むノード対を挿入することはできない。

[0071] 図 6C は、分類キー 270e “010010” の挿入後であって分類キー 270c “000111” を挿入する前の状態である分類ツリー 200c を説明する図である。上述のように、ノード 210d の下位に分類キー 270e と連結するリーフノードを含むノード対を挿入することはできないので、図 6C の点線の矢印 290e で示すように、分類キー 270e “010010” を、インデックスキー 271e “010011” に加えて、図 6B に示す

挿入位置のリーフノード 2 1 0 d のキーリンク表に連結する。この処理については後に詳細に説明するが、インデックスキーの管理領域 3 2 0 のキー分類表 3 2 1 とキーリンク表 3 2 2 の書き込みにより行われる。

[0072] 図 6 C には、分類ツリー 2 0 0 c に加えて、次に挿入される分類キーとして、分類キー 2 7 0 c “0 0 0 1 1 1” が記載され、分類キー 2 7 0 c の挿入位置としてノード 2 1 0 b が示されている。また、探索経路カウンタのカウント値は 2 段に亘って表示されている。上段に表示されたカウント値は、挿入位置を求める際にカウントされたものであり、1 段目のルートノード 2 1 0 a に対応するカウント値 1 から、2 段目の挿入位置であるノード 2 1 0 b に対応するカウント値 2 にカウントアップされている。下段に表示されたカウント値は、挿入位置であるノード 2 1 0 b を検索開始ノードとして、ノード 2 1 0 b より下位の階層のリーフノードを順次検索し、リーフノードの段数を求める際にカウントされたものであり、検索開始ノード 2 1 0 b に対応するカウント値 2 から、3 段目のリーフノード 2 1 0 d に対応するカウント値 3 にカウントアップされている。すると、ノード 2 1 0 b の下位に分類キー 2 7 0 c と連結するリーフノードを含むノード対を挿入するとリーフノード 2 1 0 d の段数は 4 となる。したがって、段数制限を超えるので、ノード 2 1 0 b の下位に分類キー 2 7 0 c と連結するリーフノードを含むノード対を挿入することはできない。

[0073] 図 6 D は、ノード対を挿入するため挿入位置の下位のリーフノードのキーリンク表に連結したインデックスキーを、挿入位置のリーフノードのキーリンク表に連結した状態の分類ツリー 2 0 0 d を説明する図である。上述のとおり、図 6 C に示す分類ツリー 2 0 0 c の挿入位置であるノード 2 1 0 b の下位にノード対を挿入することはできない。そこで、本発明によれば、ノード対を構成する 2 つのリーフノード（子ノード）の親ノードをリーフノードとし、親ノードであるリーフノードのキーリンク表に、2 つの子ノードのキーリンク表に連結するインデックスキーを連結する（以下、挿入位置の下位のリーフノードを統合する、あるいはリーフノードを統合する、のように表記

することがある。) ことにより、分類ツリーの段数を減らして挿入する分類キーと連結するリーフノードを含むノード対の挿入を可能にする。

[0074] 図6Dに示すように、分類ツリー200dのノード210bは、図6Cに示す分類ツリー200cのリーフノード210dの内容を、リーフノード210dとリーフノード211dからなるノード対201dの上位のブランチノードであるノード210bに書き込んだものである。そして、図6Dの点線の矢印290eで示すように、インデックスキー270e “010010”、インデックスキー271e “010011”、インデックスキー271d “011010”を、図6Dに示すリーフノード210bのキーリンク表に連結する。

[0075] 図6Dには、分類ツリー200dに加えて、次に挿入される分類キーとして、分類キー270c “000111”が記載され、分類キー270cの挿入位置としてノード210bが示されている。また、リーフノード210bにおける探索経路カウンタの下段のカウント値は、1つ減って、2と表示されている。したがって、分類ツリー200dの段数が1つ減り、分類ツリー200dのノード210bの下位に、分類キー270cと連結するリーフノードを含むノード対を挿入することが可能である。

[0076] 図6Eは、挿入位置の下位のリーフノードを統合した後の分類ツリー200dに分類キー270c “000111”を挿入した分類ツリー200eの状態を説明する図である。分類キー270c “000111”と連結するリーフノードを含むノード対201cは、ノード210bの下位に挿入されている。分類ツリー200dではリーフノードであったノード210bはブランチノードとなっており、弁別ビット位置には、分類キー270cとインデックスキー270eの差分ビット位置1が格納されている。ノード210bの代表ノード番号220bは挿入されたノード対201cの代表ノード210cが配置された配列要素の配列番号である。

[0077] 分類キー270cがインデックスキー270eより小さいので、分類キー270cと連結するリーフノードはノード対201cのノード[0]であるノ

ード210cとなる。その分類参照ポインタ250cには、分類参照ポインタ280cが格納されている。そして、点線の矢印290cで示すように、リーフノード210cには分類参照ポインタ280cによりインデックスキーの管理領域320を介してインデックスキー記憶領域311に記憶された分類キー270c“000111”が対応付けられている。すなわち、分類キー270c“000111”は、リーフノード210cに分類される。一方、図6Eに示すように、図6Dに示すノード210bの内容が、図6Eのノード211cにコピーされている。

以上説明した流れにより、図5に示す分類ツリー200が生成される。

[0078] 次に、図7A及び図7Bを参照して、本発明の一実施の形態におけるインデックスキーの分類処理と、分類された分類済みキーを出力先に出力する分配処理の全体の処理の概要を説明する。

[0079] 図7Aは、本発明の一実施の形態における分類処理の処理フロー例を示す図である。分類処理は、分類キーにより分類ツリーを生成するとともに、インデックスキーの管理領域にデータの書き込みを行い分類対象であるインデックスキーを分類ツリーのリーフノードと対応付けて分類するものである。なお、以下の説明においては、分類キーにより分類ツリーを生成するとともに、インデックスキーの管理領域にデータの書き込みを行うことを、単に分類ツリーを生成するということがある。

[0080] 図7Aに示すように、まずステップS701において、分類ツリーの最大段数を取得する。この最大段数の取得は、直接最大段数を取得するものであってもよいし、分類対象のキーを分類するブロック数から割り出すものであってもよい。先に述べたように、例えばブロック数が8であれば最大段数が4であるように、ブロック数より小さくない最小の2のべき乗のべき乗数に1を加えた数値を最大段数とすることができる。

[0081] 次にステップS701aにおいて、インデックスキーの記憶領域のキー参照ポインタに、インデックスキーの記憶領域の先頭のキーの記憶位置を設定する。ここで、インデックスキーの記憶領域のキー参照ポインタは、先に述べ

た、特に図示されてはいないが、処理の途中で各種の値を設定しておき、後の処理でその設定値を用いるための一時記憶領域の1つである。

[0082] 次にステップS 7 0 2において、分類対象の全てのキーは処理済みか判定し、処理済みであれば、ステップS 7 0 6を介して、図7 Bに示すステップS 7 1 1以降の処理に進み、処理済みでなければ、ステップS 7 0 3に進む。ステップS 7 0 3では、インデックスキーの記憶領域より、キー参照ポインタの指すキーを読み出し、分類キーに設定する。そしてステップS 7 0 4において、分類キーにより分類ツリーを生成する。ステップS 7 0 4の分類ツリーの生成処理の詳細については、後に図8、図9 A～図9 Cを参照して説明する。

[0083] 次にステップS 7 0 5において、インデックスキーの記憶領域のキー参照ポインタに、インデックスキーの記憶領域に記憶された次のキーの記憶位置を設定してステップS 7 0 2に戻る。ステップS 7 0 2～ステップS 7 0 5のループ処理を、ステップS 7 0 2で全てのキーは処理済みであると判定されるまで繰り返し、ステップS 7 0 2で全てのキーは処理済みであると判定される分類処理を終了し、ステップS 7 0 6において、キー分類表のキーの出力先にキーの出力先を設定して、図7 Bに示すステップS 7 1 1以下の分配処理に進む。

[0084] 図7 Bは、本発明の一実施の形態における分配処理の処理フロー例を示す図である。分配処理は、分類処理で書き込みの行われたインデックスキーの管理領域のデータにより、ブロック毎に分類された分類キーを読み出して出力先に出力するものである。

[0085] 図7 Bに示すように、まずステップS 7 1 1において、検索開始ノードの配列番号に、分類ツリーのルートノードの配列番号を設定する。次にステップS 7 1 2において、検索処理においてたどるノードの配置された配列要素の配列番号を格納する探索経路スタックに、最初に格納される配列番号として終端番号を格納する。この終端番号は検索処理でたどるノードの格納された配列要素の配列番号と区別ができるものであればよい。

- [0086] 次にステップS 7 1 4において、検索開始ノードより配列を検索し、インデックスキーの最小値を含むリーフノードを求め、リーフノードのキーリンク表に連結したインデックスキーを順次取り出す。ステップS 7 1 4の処理の詳細は、後に図 1 5 A 及び図 1 5 B を参照して説明する。
- [0087] 次にステップS 7 1 8 で、探索経路スタックから配列番号を取り出し、探索経路スタックのスタックポインタを1つ減らし、ステップS 7 1 9 に進む。ステップS 7 1 9 では、ステップS 7 1 8 で取り出した配列番号が終端番号であるか判定する。判定の結果、終端番号であれば、処理を終了し、終端番号でなければ、ステップS 7 2 1 に進む。
- [0088] ステップS 7 2 1 では、ステップS 7 1 8 で取り出した配列番号から、その配列番号のノードがノード対のどちらの配列要素に格納されているかのノード位置を得る。例えばノード [0] については偶数番号の配列の配列要素に格納する等により、配列番号からノード位置を求めることができる。
- [0089] そして、ステップS 7 2 2 で、ステップS 7 2 1 で得たノード位置がノード [1] 側であるか否かを判定する。ステップS 7 2 2 においてノード [1] 側と判定された場合は、ステップS 7 1 8 に戻る。
- [0090] ステップS 7 2 2 においてノード [0] 側と判定されると、ステップS 7 2 3 に進み、配列番号に1を加算し、そのノードと対をなすノード [1] の配列番号を得る。そして、ステップS 7 2 4 で、検索開始ノードの配列番号に、ステップS 7 2 3 で得たノード [1] の配列番号を設定してステップS 7 1 4 に戻る。
- [0091] 上述のステップS 7 1 4 ~ステップS 7 2 4 のループ処理を、ステップS 7 1 8 で探索経路スタックのスタックポインタを1つ減らしながら、ステップS 7 1 9 で探索経路スタックから取り出した配列番号が終端番号であると判定されるまで繰り返す。探索経路スタックから取り出した配列番号が終端番号であれば、分類ツリーの全てのリーフノードについての処理が完了したので、処理を終了する。
- [0092] 上述の図 7 B に示す分類ツリーに格納された分類キーを取り出す処理は、

先に述べた特許文献４の図面第１０図に開示された、カップルドノードツリーからインデックスキーを昇順で取り出す処理（以下、先発明の処理ということがある）と類似したものである。図７Ｂに例示したフローでは、リーフノードを昇順で求めているが、降順で求めることによっても、各ブロックに分類されたキーを取り出せることは、当業者に明らかである。

なお、先発明ではリーフノードと連結するインデックスキーは１つであるのに対して、本発明ではリーフノードのキーリンク表に連結するインデックスキーは複数でありえるので、ステップＳ７１４において、リーフノードのキーリンク表に連結するインデックスキーを順次取り出す点で本質的に異なる。

[0093] 次に、図８、図９Ａ～図９Ｃを参照して、図７Ａに示すステップＳ７０４の分類キーにより、分類ツリーを生成する処理の詳細を説明する。

図８は、本発明の一実施の形態における分類キーにより分類ツリーを生成する全体の処理フロー例を説明する図である。図８に示す処理の前提として、図７Ａに示すステップＳ７０３において、分類キーが一時記憶領域に設定されている。

[0094] 図８に示すように、まずステップＳ８０１において、ルートノードの配列番号は登録済みか判定する。ルートノードの配列番号が登録済みでなければ、ステップＳ８０２以下の処理に進み、上記一時記憶領域に設定されている分類キーと連結するリーフノードをルートノードとする分類ツリーが新たに生成される。

[0095] ステップＳ８０２では、インデックスキーの管理領域の分類参照ポインタとキー管理ポインタを取得する。ここで取得される分類参照ポインタとキー管理ポインタは、インデックスキーの管理領域３２０をデータ格納装置３０８上に設定し、最初に用いるキー分類表のエントリとキーリンク表のエントリのアドレスをそれぞれ決めることにより得られるものである。

[0096] 次にステップＳ８０３において、分類参照ポインタの指すキー分類表の先頭リンクと末尾リンクに、キー管理ポインタを、最小値キーと最大値キーに

分類キーを書き込み、ステップS 8 0 4において、キー管理ポインタの指すキーリンク表のキー参照ポインタに、図7 AのステップS 7 0 3でキーの読み出しに用いたキー参照ポインタを書き込む。

例えば、最初に処理される分類キーが、図2 Bに例示する分類キー2 7 1 dで、図5に示すキー分類表3 2 1の最初に用いるエントリを指す分類ポインタが2 8 0 d、キーリンク表3 2 2の最初に用いるエントリを指すキー管理ポインタが3 7 1 dであるとする、分類ポインタ2 8 0 dの指すキー分類表3 2 1の最小値キー3 1 2 aと最大値キー3 1 2 bに分類キー2 7 1 dが、先頭リンク3 1 2 cと末尾リンク3 1 2 dにキー管理ポインタ3 7 1 dが、それぞれ書き込まれ、キー管理ポインタ3 7 1 dの指すキーリンク表3 2 2のキー参照ポインタ3 1 3 aに図2 Bに示す参照ポインタ4 8 1 dが、書き込まれる。

[0097] 次にステップS 8 0 5において、配列から空きのノード対を求め、そのノード対のうち代表ノードとなるべき配列要素の配列番号を取得し、ステップS 8 0 6において、ステップS 8 0 5で得た配列番号に0を加えた配列番号を求める。（実際には、ステップS 8 0 5で取得した配列番号に等しい。）。

[0098] さらにステップS 8 0 7において、ステップS 8 0 6で得た配列番号の配列要素に、挿入するルートノードのノード種別に1（リーフノード）と分類参照ポインタにステップS 8 0 2で得た分類参照ポインタを書き込み、ステップS 8 0 8で、ステップS 8 0 5で取得したルートノードの配列番号を登録して処理を終了する。

上述の最初に処理される分類キーがキー2 7 1 dとすると、図6 Aに示すリーフノード2 1 0 bが、最初に生成される分類キーとなる。なお、図5に示すキー分類表には分類参照ポインタ2 8 0 dの指すエントリは記載されていないが、分類キーを順次挿入していく過程において、当初用いられていたキー分類表のエントリが他のエントリと結合されて用いられなくなる場合がある。

[0099] 一方、ステップS 8 0 1において、ルートノードの配列番号が登録済みで

あると判定されると、ステップS 8 0 9に進み、ルートノードが登録済みである分類ツリーに分類キーを挿入することにより、新たな分類ツリーを生成して処理を終了する。ステップS 8 0 9の処理の詳細は、次に図9 A～図9 Cを参照して説明する。

[0100] 図9 Aは、本発明の一実施の形態における分類キーを分類ツリーに挿入する処理の初段の処理フロー例を示す図である。この初段の処理は、分類ツリーのルートノードを検索開始ノードとし、分類キーを検索キーとしてリーフノードを求める検索処理である。先に図6 Aを参照して説明したように、リーフノードに至る探索経路上のノードの階層に対応して探索経路カウンタのカウント値がカウントアップされる。

[0101] 図9 Aに示すように、まずステップS 9 0 1で、配列番号に、ルートノードの配列番号を設定し、ステップS 9 0 3に進む。

[0102] ステップS 9 0 3では、配列から、配列番号の指す配列要素をノードとして読み出し、ステップS 9 0 4でノードから、ノード種別を取り出し、ステップS 9 0 5でノード種別はブランチノードであるか否かを判定する。

[0103] ステップS 9 0 5の判定において、読み出したノードがブランチノードと判定された場合は、ステップS 9 0 6に進み、ノードから弁別ビット位置を取り出し、更に、ステップS 9 0 7で、取り出した弁別ビット位置に対応するビット値を分類キーから取り出す。そしてステップS 9 0 8において、ノードから代表ノード番号を取り出す。さらにステップS 9 0 9において、代表ノード番号に分類キーから取り出したビット値を加えて新たな配列番号を得て、ステップS 9 0 2に戻る。

[0104] 以降、ステップS 9 0 5の判定においてリーフノードと判定されてステップS 9 1 0に進むまで、ステップS 9 0 3～ステップS 9 0 9のループ処理を繰り返す。ステップS 9 1 0では、リーフノードから分類参照ポインタを取り出し、図9 Bに示すステップS 9 1 1に進む。

[0105] 図9 Bは、本発明の一実施の形態における分類キーを分類ツリーに挿入する処理の中段の処理フロー例を示す図である。この中段の処理は、初段の処理

で求められたリーフノードのキーリンク表に連結しているキーと分類キーの値を比較し、分類キーをリーフノードのキーリンク表に連結するかを決定し、連結する場合は連結し、連結しない場合には、分類キーと連結するリーフノードを含むノード対の挿入位置を求める処理の準備を行うものである。

[0106] 図9Bに示すように、まずステップS911において、図9AのステップS910で取り出した分類参照ポインタの指すキー分類表の最小値キーと最大値キーを読み出す。次にステップS915に進み、分類キーは最小値キーより小さいか判定する。

[0107] 分類キーが最小値キーより小さくなければ、ステップS916において、さらに分類キーは最大値キーより大きいと判定する。ステップS916において、分類キーは最大値キーより大きくないと判定される、すなわち、分類キーの値が最小値キーと最大値キーの値の間であると判定されると、ステップS917で、分類キーを、リーフノードのキーリンク表に連結して挿入処理を終了する。ステップS917の処理の詳細は、後に図11を参照して説明する。

[0108] 一方、ステップS916において、分類キーは最大値キーより大きいと判定されると、ステップS918に進み、インデックスキーに最大値を設定してステップS920に進む。

また、先に述べたステップS915において、分類キーは最小値キーより小さいと判定されると、ステップS919において、インデックスキーに最小値を設定してステップS920に進む。

[0109] ステップS920においては、分類キーとステップS918またはステップS919で設定されたインデックスキーとのビット列比較を例えば排他的論理和で行い、差分ビット列を得る。次にステップS921において、ステップS920で得た差分ビット列から、上位0ビット目から見た最初の不一致ビットのビット位置（差分ビット位置）を得る。この処理は、例えばプライオリティエンコーダを有するCPUではそこに差分ビット列を入力し、不一致のビット位置を得ることができる。また、ソフト的にプライオリティエン

コーダと同等の処理を行い最初の不一致ビットのビット位置を得ることも可能である。

[0110] 次にステップS 9 2 2 aにおいて、挿入位置の配列番号に、ルートノードの配列番号を設定し、図9 Cに示すステップS 9 2 2 bに進む。

[0111] 図9 Cは、本発明の一実施の形態における分類キーを分類ツリーに挿入する処理の後段の処理フロー例を示す図である。この後段の処理は、図9 Bに示す中段の処理においてなされた準備に基づいて、分類キーをリーフノードのキーリンク表に連結するものである。

[0112] 図9 Cに示すように、まずステップS 9 2 2 bにおいて、探索経路カウンタに、初期値として0を設定してステップS 9 2 3に進む。

[0113] ステップS 9 2 3では、探索経路カウンタのカウント値を1つ増やす。次にステップS 9 2 4 aにおいて、配列から、挿入位置の配列番号の指す配列要素をノードとして読み出し、ステップS 9 2 4 bにおいて、ノードから、ノード種別を取り出してステップS 9 2 5に進む。

[0114] ステップS 9 2 5では、ステップS 9 2 4 bで取り出したノード種別はブランチノードを示すか判定する。ノード種別がブランチノードを示すものではない、すなわちリーフノードを示すものであれば、ステップS 9 3 2に進む。

[0115] 一方、ノード種別がブランチノードを示すものであればステップS 9 2 6に進み、ノードから弁別ビット位置を取り出し、ステップS 9 2 7において、弁別ビット位置はステップS 9 2 1で得た差分ビット位置より上位の位置関係か判定する。弁別ビット位置が差分ビット位置より上位でなければ、ステップS 9 3 5に進む。

[0116] 一方、弁別ビット位置が差分ビット位置より上位であれば、ステップS 9 2 8 aに進む。ステップS 9 2 8 aでは、分類キーから、弁別ビット位置の指すビット値を取り出し、ステップS 9 2 8 bでノードから、代表ノード番号を取り出す。そして、ステップS 9 2 9において、その代表ノード番号にステップS 9 2 8 aで得た値を加えた値を、挿入位置の配列番号に設定してス

テップS 9 2 3に戻る。

[0117] 上述のステップS 9 2 3～ステップS 9 2 9のループ処理を、ステップS 9 2 5でノード種別がリーフノードのものと判定されるか、あるいはステップS 9 2 7でブランチノードの弁別ビット位置が差分ビット位置より上位でないと判定されるまで、ルートノードからの探索を繰り返す。ステップS 9 2 5あるいはステップS 9 2 7でループ処理から抜け出す直前のステップS 9 2 9で設定された挿入位置の配列番号あるいはルートノードがリーフノードの場合はステップS 9 2 2 aで設定された挿入位置の配列番号が、分類キーと連結するリーフノードを含むノード対の挿入位置を示すものである。

[0118] 上述のステップS 9 2 3～ステップS 9 2 9のループ処理は、探索経路上のブランチノードをたどって、その弁別ビット位置と、分類キーとインデックスキーの差分ビット位置との相対的位置関係を判定するために、図9 Aに示す検索処理と同様に、検索開始ノードであるルートノードからリーフノードに向かって探索経路を再度たどるものである。

[0119] 上記ステップS 9 2 5で、ステップS 9 2 4で取り出したノード種別はリーフノードを示すものであると判定されると、ステップS 9 3 2に進み、探索経路カウンタは分類ツリーの最大段数の値である最大値を示しているか判定する。

[0120] 探索経路カウンタのカウント値は最大値ではないと判定されると、ステップS 9 3 3に進み、分類キーを、挿入位置に挿入して挿入処理を終了する。挿入位置のノードがリーフノードであって探索経路カウンタのカウント値は最大値ではない状態の一例は、先に図6 Aを参照して説明した状態である。そして、分類キーを、挿入位置に挿入した状態の一例は、図6 Bに示すものである。

ステップS 9 3 3の処理の詳細は、後に図10を参照して説明する。

[0121] 一方、ステップS 9 3 2で、探索経路カウンタのカウント値が最大値であると判定されると、ステップS 9 3 4に進み、分類キーを、リーフノードのキーリンク表に連結して挿入処理を終了する。この探索経路カウンタのカウ

ント値が最大値である状態の一例は、先に図 6 B を参照して説明した状態である。そして、分類キーを、リーフノードのキーリンク表に連結した状態の一例は、図 6 C に示すものである。

なお、リーフノードの直近上位のブランチノードの弁別ビット位置までの上位のビット値は、同一のブロックに分類されるどのキーでも一致し、リーフノードの直近上位のブランチノードの弁別ビット位置までに異なるビット値のあるキーは、他のブロックに分類されることから、ステップ S 9 3 4 において分類キーをリーフノードに連結しても、各ブロックに分類された分類キーの値の範囲に重複が生じることはない。

ステップ S 9 3 4 の処理の詳細は、後に図 1 1 を参照して説明する。

- [0122] 先に述べたステップ S 9 2 7 において、弁別ビット位置はステップ S 9 2 1 で得た差分ビット位置より上位の位置関係ではないと判定されると、ステップ S 9 3 5 に進む。
- [0123] ステップ S 9 3 5 では、分類キーを挿入位置に挿入したときに、分類ツリーの段数が制限を超えないことを保証する処理を行う。すなわち、挿入位置に分類キーを挿入したときに、挿入位置より下位のリーフノードの段数が最大値を越さないかチェックを行い、制限を越すリーフノードがあれば、そのリーフノードを含むノード対の親ノードをリーフノードとし、該ノード対を構成するリーフノードのキーリンク表に連結するキーを、リーフノードとした親ノードのキーリンク表に連結することを、挿入位置より下位のリーフノードのすべてについて行う。このステップ S 9 3 5 の処理により、挿入位置に分類キーを挿入しても、分類ツリーの段数が最大値を越えることはない。
- [0124] 挿入位置に分類キーを挿入したときに、挿入位置より下位のリーフノードの段数が最大値を越える状態の一例は、先に図 6 C を参照して説明した状態である。図 6 C に示す状態では、挿入位置の下位のリーフノード 2 1 0 d における探索経路カウンタの値は最大値の 3 となっている。
- [0125] そして、リーフノード 2 1 0 d を含むノード対 2 0 1 d の親ノード 2 1 0 b をリーフノードとして、リーフノード 2 1 0 d のキーリンク表に連結するキ

ー270e、271eとリーフノード210dを含むノード対201dのも
う一方のリーフノード211dのキーリンク表に連結するキー271dとを
、リーフノードとした親ノード210bのキーリンク表に連結した状態の一
例は、図6Dに示すものである。

ステップS935の詳細は、後に図12を参照して説明する。

[0126] ステップS935に続いて、ステップS936に進み、先に述べたステップ
S933と同じく、分類キーを挿入位置に挿入して挿入処理を終了する。リ
ーフノードを含むノード対の親ノードをリーフノードとし、該ノード対を構
成するリーフノードのキーリンク表に連結するキーをリーフノードとした親
ノードのキーリンク表に連結することにより、探索経路カウンタの値を最大
値から1つ減らし、分類キーを挿入位置に挿入した状態の一例は、先に図6
Eを参照して説明した状態である。

ステップS936の処理の詳細は、先にステップS933について述べたよ
うに、次に図10を参照して説明する。

[0127] 図10は、本発明の一実施の形態における分類キーを挿入位置に挿入する処
理フロー例を示す図であり、図9Cに示すステップS933及びステップS
936の詳細な処理を説明するものである。

[0128] 図10に示すように、まずステップS1001において、配列から空きのノ
ード対を求め、そのノード対のうち代表ノードとなるべき配列要素の配列番
号を取得する。次にステップS1002に進み、分類キーと図9Bに示す処
理で設定したインデックスキーの大小を比較し、分類キーが大きいときは値
1を小さいときは値0のブール値を得る。

[0129] さらに、ステップS1003に進み、ステップS1001で得た代表ノード
の配列番号にステップS1002で得たブール値を加算した配列番号を得る
。また、ステップS1004において、ステップS1001で得た代表ノード
の配列番号にステップS1002で得たブール値の論理否定値を加算した
配列番号を得る。

[0130] 次にステップS1005に進み、インデックスキーの管理領域の分類参照

ポインタとキー管理ポインタを取得する。ここでインデックスキーの管理領域の分類参照ポインタとキー管理ポインタを取得するのは、挿入する分類キーを含むリーフノードに対応するキー分類表とキーリンク表のエントリを確保するためである。

[0131] キーリンク表のエントリ数は分類対象のキーの数であるから、分類対象のキーの数だけ空きエントリの領域を予めインデックスキーの管理領域に確保しておき、取得要求があるたびに順次空きエントリの実アドレスをキー管理ポインタとして渡すことにより、キー管理ポインタの取得を実行することができる。

[0132] キー分類表のエントリ数は、最終的に必要な数はキーを分類するブロック数、言い換えれば分類ツリーのリーフノードの数であるが、キーリンク表と同様に分類対象のキーの数だけ空きエントリの領域を予めインデックスキーの管理領域に確保しておき、取得要求があるたびに順次空きエントリの実アドレスを分類参照ポインタとして渡すことにより、分類参照ポインタの取得を実行することができる。

[0133] 先に図9Cに示すステップS935の説明において述べた、リーフノードを含むノード対の親ノードをリーフノードとし、該ノード対を構成するリーフノードのキーリンク表に連結するキーを、リーフノードとした親ノードのキーリンク表に連結する処理によって不要となったエントリは、削除されることになる。

[0134] 次にステップS1006において、ステップS1005で取得した分類参照ポインタの指すキー分類表の先頭リンクと末尾リンクにステップS1005で取得したキー管理ポインタを、最小値キーと最大値キーに分類キーを書き込み、ステップS1007において、キー管理ポインタの指すキーリンク表のキー参照ポインタに、インデックスキーの記憶領域のキー参照ポインタを書き込む。インデックスキーの記憶領域のキー参照ポインタは、図7Aに示すステップS701あるいはステップS705で設定されたものである。

[0135] 次にステップS1008において、ステップS1003で得た配列番号の

指す配列要素のノード種別に1（リーフノード）を、分類参照ポインタにステップS1005で得た分類参照ポインタを書き込む。

[0136] ステップS1009に進み、配列から挿入位置の配列番号の配列要素の内容を読み出し、ステップS1010において、ステップS1004で得た配列番号の指す配列要素に、ステップS1009で読み出した内容を書き込む。ここで、挿入位置の配列番号は、図9Cに示すステップS929で設定されたものである。

[0137] 最後にステップS1011において、挿入位置の配列番号の指す配列要素のノード種別に0（ブランチノード）を、弁別ビット位置に図9Cに示すステップS921で得た差分ビット位置を、代表ノード番号にステップS1001で得た代表ノード番号を書き込み、処理を終了する。

[0138] 以上説明した図10に示す処理フロー例に、先に説明した図6A及び図6Bの例示を対応付けると、ステップS1001で得た代表ノード番号は図6Bに示すノード210bの代表ノード番号220b、ステップS1002で得たブール値は0、ステップS1008で生成したリーフノードは図6Bに示すノード210d、ステップS1009で読み出した配列要素の内容は図6Aに示すノード210b、ステップS1010で生成したノードは図6Bに示すノード211d、ステップS1011で生成したブランチノードは図6Bに示すノード210bである。

[0139] 図11は、本発明の一実施の形態における、分類キーをリーフノードのキーリンク表に連結する処理フロー例を示す図であり、図9Bに示すステップS917及び図9Cに示すステップS934の詳細な処理を説明するものである。分類キーをリーフノードのキーリンク表に連結する処理は、分類ツリーには変更を加えることなく、インデックスキーの管理領域のキー分類表とキーリンク表を書き換えることにより行われる。

[0140] 図11に示すように、まずステップS1101において、インデックスキーの管理領域のキー管理ポインタを取得する。次にステップS1102に進み、キー管理ポインタの指すキーリンク表のキー参照ポインタに、インデッ

クスキーの記憶領域のキー参照ポインタを書き込む。

[0141] 次にステップS 1 1 0 3に進み、図9 Aに示すステップS 9 1 0において取り出した分類参照ポインタの指すキー分類表の末尾リンクを読み出し、ステップS 1 1 0 4において、末尾リンクの指すキーリンク表のリンクに、ステップS 1 1 0 1で取得したキー管理ポインタを書き込む。そして、ステップS 1 1 0 5に進み、さらにキー管理ポインタを、図9 Aに示すステップS 9 1 0において取り出した分類参照ポインタの指すキー分類表の末尾リンクに書き込む。

[0142] 次にステップS 1 1 0 6において、図9 Aに示すステップS 9 1 0において取り出した分類参照ポインタの指すキー分類表の最小値キーと最大値キーを読み出し、ステップS 1 1 0 7において、分類キーは最小値キーより小さいか判定する。

[0143] 分類キーが最小値キーより小さければ、ステップS 1 1 0 8において、図9 Aに示すステップS 9 1 0において取り出した分類参照ポインタの指すキー分類表の最小値キーに分類キーを書き込み、処理を終了する。

[0144] 分類キーが最小値キーより小さくなければ、ステップS 1 1 0 9において、分類キーは最大値キーより大きいと判定する。分類キーが最大値キーより大きくなければ処理を終了し、分類キーが最大値キーより大きければ、ステップS 1 1 1 0において、図9 Aに示すステップS 9 1 0において取り出した分類参照ポインタの指すキー分類表の最大値キーに分類キーを書き込み、処理を終了する。

なお、上述の図1 1に例示する処理フローは、図9 Bに示すステップS 9 1 7と図9 Cに示すステップS 9 3 4の処理に共通のものとしているが、図9 Bに示すステップS 9 1 7の処理は分類キーの値が最小値キーの値と最大値キーの値の範囲内であることを前提としているので、図9 Bに示すステップS 9 1 7の処理としては、ステップS 1 1 0 1～ステップS 1 1 0 5の処理とすることもできる。

[0145] 図1 2は、本発明の一実施の形態における分類キーを挿入位置に挿入したと

きに、分類ツリーの段数が制限を超えないための準備を行う処理フロー例を説明する図であり、図 9 C に示すステップ S 9 3 5 の詳細な処理を説明するものである。先に図 9 C を参照して説明したように、図 1 2 に示す処理は、探索経路上の、ステップ S 9 2 9 で設定された挿入位置の配列番号の指す配列要素に配置されたブランチノードの弁別ビット位置が、ステップ S 9 2 1 で得た差分ビット位置より上位の位置関係ではないと判定されるときに実行される。

[0146] そして、挿入位置に分類キーを挿入したときに、挿入位置より下位のリーフノードの段数が最大値を越さないかチェックを行い、制限を越すリーフノードがあれば、そのリーフノードを含むノード対の親ノードをリーフノードとして、該親ノードのキーリンク表に、該ノード対を構成するリーフノードのキーリンク表に連結しているキーを連結することを、挿入位置より下位のリーフノードの全てについて行う。

[0147] そこで、図 1 2 に示す処理においては、挿入位置より下位のリーフノードを全て巡回するために、先に図 7 B を参照して説明した分類ツリーに格納された分類キーを取り出す処理と同様に、本願と同一の出願人の出願に係る、特許第 4 2 7 1 2 2 7 号の図面第 1 0 図に開示された、カップルドノードツリーからインデックスキーを昇順で取り出す処理と類似した処理を行う。

[0148] 図 1 2 に示すように、まずステップ S 1 2 0 1 において、検索開始ノードの配列番号に、挿入位置の配列番号を設定し、ステップ S 1 2 0 4 に進む。ステップ S 1 2 0 4 においては、検索開始ノードより配列を検索し、インデックスキーの最小値を含むリーフノードを求める。ここで、挿入位置の配列番号は、図 9 C に示すステップ S 9 2 9 で設定されたものである。ステップ S 1 2 0 4 の処理の詳細は、後に図 1 3 を参照して説明する。

[0149] 次にステップ S 1 2 0 5 において、探索経路カウンタのカウント値は最大値か判定する。ここで探索経路カウンタのカウント値は、ステップ S 1 2 0 4 でリーフノードを求める際にカウントされるものであり、そこで求められたリーフノードが分類ツリーの何段目にあるかを示すものである。

[0150] ステップS 1 2 0 5において、探索経路カウンタのカウント値が最大値ではないと判定されると、ステップS 1 2 0 6に進み、探索経路スタックから配列番号を取り出し、探索経路スタックのスタックポインタを1つ減らし、ステップS 1 2 0 9に進む。ステップS 1 2 0 6で探索経路スタックから取り出される配列番号は、ステップS 1 2 0 4において求めた、インデックスキーの最小値を含むリーフノードが配置された配列要素の配列番号か、ステップS 1 2 0 6～ステップS 1 2 1 2 aのループ処理の1回前のループにおけるステップS 1 2 0 6で1つ減らしたスタックポインタの指す配列番号である。

[0151] 一方、ステップS 1 2 0 5において、探索経路カウンタのカウント値は最大値であると判定されると、ステップS 1 2 0 7に分岐する。ステップS 1 2 0 7では、探索経路スタックから、探索経路スタックのスタックポインタの値を1つ減らして配列番号を取り出し、取り出した配列番号を親ノードの配列番号に設定する。ここでの親ノードは、ステップS 1 2 0 4において求めた、インデックスキーの最小値を含むリーフノードの直近上位のノードである。あるノードの直近上位のノードをそのノードの親ノードといい、直近下位のノードを子ノードという。

[0152] 次にステップS 1 2 0 8において、ステップS 1 2 0 4で求めたリーフノードのキーリンク表に連結したキーと、そのリーフノードと対をなすリーフノードのキーリンク表に連結したキーとに連結するリーフノードを、ステップS 1 2 0 6で設定した配列番号、すなわち親ノードの配列要素に書き込む。言い換えれば、ステップS 1 2 0 4で求めたリーフノードの直近上位のブランチノード（親ノード）をリーフノードとし、リーフノードとした親ノードのキーリンク表に、ステップS 1 2 0 4で求めたリーフノードのキーリンク表に連結したキーと、そのリーフノードと対をなすリーフノードのキーリンク表に連結したキーを連結する。

ステップS 1 2 0 8の処理の詳細は、後に図 1 4 を参照して説明する。

[0153] 次にステップS 1 2 0 8 aにおいて、探索経路カウンタのカウント値を1

つ減らし、ステップS 1 2 0 9に進む。ここで探索経路カウンタのカウント値を1つ減らすのは、上記ステップS 1 2 0 7の処理で親ノードをリーフノードとしたことにより、リーフノードの段数が1つ減ったからである。この処理により、再度ステップS 1 2 0 4において最小値検索を行うときの探索経路カウンタのカウント値を、探索経路の段数と一致させることができる。

[0154] ステップS 1 2 0 9では、ステップS 1 2 0 6で取り出した配列番号あるいはステップS 1 2 0 7が挿入位置の配列番号であるか判定する。判定の結果、挿入位置の配列番号であれば、挿入位置のノードより下位の全てのリーフノードについての処理が完了しているので、連結処理を終了する。判定の結果、挿入位置の配列番号でなければ、ステップS 1 2 1 1に進む。

[0155] ステップS 1 2 1 1では、ステップS 1 2 0 6あるいはステップS 1 2 0 7で取り出した配列番号から、その配列番号のノードがノード対のどちらの配列要素に格納されているかのノード位置を得る。例えばノード [0] については偶数番号の配列の配列要素に格納する等により、配列番号からノード位置を求めることができる。

[0156] そして、ステップS 1 2 1 2で、ステップS 1 2 1 1で得たノード位置がノード [1] 側であるか否かを判定する。ステップS 1 2 1 2においてノード [1] 側と判定された場合は、ステップS 1 2 1 2 aにおいて探索経路カウンタのカウント値を1つ減らしてステップS 1 2 0 6に戻る。

[0157] ステップS 1 2 1 2においてノード [0] 側と判定されると、ステップS 1 2 1 3に進み、配列番号に1を加算し、そのノードと対をなすノード [1] の配列番号を得る。そして、ステップS 1 2 1 4で、検索開始ノードの配列番号に、ステップS 1 2 1 3で得たノード [1] の配列番号を設定してステップS 1 2 0 4に戻る。

[0158] 上述のステップS 1 2 0 4～ステップS 1 2 1 4のループ処理を、ステップS 1 2 0 6で探索経路スタックのスタックポインタを1つ減らしながら、あるいはステップS 1 2 0 7～ステップS 1 2 0 8 aの処理で親ノードをリーフノードとすることによりリーフノードの段数を減らしながら、ステップ

S 1 2 0 9で探索経路スタックから取り出した配列番号が挿入位置の配列番号であると判定されるまで繰り返す。

[0159] 図 6 Cに示す例のように、挿入位置のノードの直近下位のノードがリーフノードでその段数が最大値の場合、ステップS 1 2 0 4で、最小値を含むノードとしてノード2 1 0 dが得られ、探索経路スタックには、挿入位置のノードの配列番号である2 2 0 aとノード2 1 0 dが配置された配列要素の配列番号2 2 0 bが格納される。探索経路カウンタのカウント値は最大値となっているので、ステップS 1 2 0 7では配列番号2 2 0 aが取り出され、ステップS 1 2 0 8、ステップS 1 2 0 8 aの処理に続いてステップS 1 2 0 9の判定において、配列番号は挿入位置の配列番号であると判定されて処理は終了する。

[0160] 仮に、分類ツリーの最大段数が4であると図 6 Cの例を変更すると、ステップS 1 2 0 5の判定によりステップS 1 2 0 6に進み、配列番号2 2 0 bが探索経路スタックから取り出され、ステップS 1 2 0 9の判定は否定的なものとなり、ステップS 1 2 1 1、ステップS 1 2 1 2及びステップS 1 2 1 3を介してステップS 1 2 1 4に至る。ステップS 1 2 1 4において、検索開始ノードの配列番号に配列番号2 2 0 b + 1が設定され、ノード2 1 1 dを検索開始ノードとしてステップS 1 2 0 4の最小値検索が行われ、インデックスキーの最小値を含むリーフノードとしてノード2 1 1 dが得られる。そして、ステップS 1 2 0 6～ステップS 1 2 1 2 aのループ処理が実行され、ステップS 1 2 0 6で挿入位置の配列番号2 2 0 aが取り出されるとステップS 1 2 0 9の判定により処理を終了する。

[0161] 最初の最小値検索が行われるときは、挿入位置のノードをルートノードとする部分木の最小値を含むリーフノードが探索されることから、そのリーフノードのノード位置はノード[0]側であり、同一ノード対を構成するノード[1]を検索開始ノードとして次の最小値検索が行われる。最後には、挿入位置のノードをルートノードとする部分木の最大値を含むリーフノードが探索される。そのリーフノードのノード位置はノード[1]側であり、ステップ

S 1 2 0 6において探索経路スタックから挿入位置の配列番号が取り出されるまで、ステップS 1 2 0 6～ステップS 1 2 1 2 aのループ処理が繰り返され、ステップS 1 2 0 9で配列番号は挿入位置の配列番号であると判定されて処理を終了する。

[0162] 図 1 3 は、本発明の一実施の形態におけるインデックスキーの最小値を含むノードを検索する処理フロー例を説明する図であり、図 1 2 に示すステップ S 1 2 0 4 の詳細な処理を説明するものである。

[0163] 図 1 3 に、示すように、まずステップS 1 3 0 1において、配列番号に、検索開始ノードの配列番号を設定する。検索開始ノードの配列番号は、図 1 2 に示すステップS 1 2 0 1、あるいはステップS 1 2 1 4で設定されている。

[0164] 次にステップS 1 3 0 2において、探索経路スタックに配列番号を格納する。そして、ステップS 1 3 0 3で、配列から配列番号の指す配列要素をノードとして読み出し、ステップS 1 3 0 4で、該読み出したノードからノード種別を取り出してステップS 1 3 0 5に進む。

[0165] ステップS 1 3 0 5では、ノード種別はブランチノードのものか判定し、ノード種別がブランチノードのものと判定されると、ステップS 1 3 0 5 aに進み、探索経路カウンタの値を1つ増やす。次にステップS 1 3 0 6に進み、ノードから配列の代表ノード番号を取り出し、ステップS 1 3 0 7で、取り出した代表ノード番号に値0を加算してその結果を新たな配列番号として、ステップS 1 3 0 2に戻る。

[0166] 以降、ステップS 1 3 0 5において、ステップS 1 3 0 4で取り出されたノードがリーフノードと判定されるまでステップS 1 3 0 2からステップS 1 3 0 7までの処理を繰り返し、ステップS 1 3 0 5で、ステップS 1 3 0 4で取り出されたノードがリーフノードと判定されると処理を終了する。

[0167] なお、上述の図 1 3 を参照して説明したインデックスキーの最小値を含むリーフノードを検索する処理では、カップルドノードツリーは配列に記憶されたものとして説明したが、カップルドノードツリーが配列に記憶されるこ

とは必須ではなく、ノード対を構成する2つのノードのうちの代表ノードのみあるいは代表ノードと隣接した記憶領域に配置されたノードのみをリンクしてリーフノードに至ることによりインデックスキーの最小値を含むリーフノードの検索が可能であることは明らかである。

[0168] 図14は、本発明の一実施の形態におけるリーフノードを含むノード対の親ノードをリーフノードとして、該親ノードのキーリンク表に該ノード対のキーリンク表を連結する処理フロー例を説明する図であり、図12に示すステップS1208の処理の詳細を説明するものである。リーフノードを含むノード対のキーリンク表を親ノードのキーリンク表に連結する処理においては、ノード対をなす各ノードに対応する分類参照ポインタは親ノードの分類参照ポインタに統合される。

[0169] 図14に示すように、まずステップS1401において、配列から、親ノードの配列番号の指す配列要素をノードとして読み出し、ステップS1402で、該読み出したノードから、代表ノード番号を取り出す。図6Cの例示では、親ノード210bの代表ノード番号220bが取り出される。

[0170] 次にステップS1403に進み、代表ノード番号に値1を加えて、1側の配列番号を得る。そしてステップS1404において、配列から、1側の配列番号の指す配列要素をノードとして読み出し、ステップS1405において、ノードから、分類参照ポインタを取り出し、1側の分類参照ポインタに設定する。図6Cの例示では、配列番号220b+1の指すノード211dが読み出され、分類参照ポインタ280dが、一時的記憶領域である1側の分類参照ポインタに設定される。

[0171] 次にステップS1406に進み、代表ノード番号に値0を加えて、ノード位置0の配列番号を得る。そしてステップS1407において、配列から、0側の配列番号の指す配列要素をノードとして読み出し、ステップS1408において、ノードから、分類参照ポインタを取り出し、0側の分類参照ポインタに設定する。図6Cの例示では、配列番号220bの指すノード210dが読み出され、分類参照ポインタ280eが、一時的記憶領域である0側

の分類参照ポインタに設定される。

- [0172] 次にステップS 1 4 0 9において、1側の分類参照ポインタの指すキー分類表の、先頭リンクと末尾リンクと最大値キーを読み出し、1側の先頭リンクと1側の末尾リンクに設定するとともに、最大値キーを1側の最大値キーに設定する。
- [0173] 次にステップS 1 4 1 0において、0側の分類参照ポインタの指すキー分類表の末尾リンクを読み出し、ステップS 1 4 1 1において、該読み出した末尾リンクの指すキーリンク表のリンクに、ステップS 1 4 0 9で設定した1側の先頭リンクを書き込む。そしてステップS 1 4 1 2において、0側の分類参照ポインタの指すキー分類表の末尾リンクに1側の末尾リンクを、最大値キーに1側の最大値キーを書き込む。
- [0174] 以上の処理により、リーフノードとする親ノードの分類参照ポインタを0側の子ノードの分類参照ポインタとし、それに合わせたキーリンク表とキー分類表の書き換えが行われる。上記ステップS 1 4 1 1の、0側の分類参照ポインタの指すキー分類表の末尾リンクの指すキーリンク表のリンクに、1側の先頭リンクを書き込む処理は、0側のリーフノードのキーリンク表に連結したキーに続いて1側のリーフノードのキーリンク表に連結したキーへのリンクを設定するものである。
- [0175] 次にステップS 1 4 1 3に進み、親ノードの配列番号の指す配列要素に、ステップS 1 4 0 7で読み出した内容を書き込む。図6 C及び図6 Dに示す例示では、図6 Cに示すノード2 1 0 dの内容が、図6 Dに示すノード2 1 0 bに書き込まれる。
- [0176] 最後に、ステップS 1 4 1 4で、S 1 4 0 1で得た代表ノード番号の指すノード対を削除し、ステップS 1 4 1 5で、S 1 4 0 5で得た1側の分類参照ポインタの指すキー分類表のエントリを削除して処理を終了する。
- [0177] 次に、図1 5 A及び図1 5 Bを参照して、検索開始ノードより配列を検索し、インデックスキーの最小値を含むリーフノードを求め、リーフノードのキーリンク表に連結したインデックスキーを順次取り出す処理について説明す

る。図 1 5 A 及び図 1 5 B に例示する処理フローは、図 7 B に示すステップ S 7 1 4 の処理の詳細を説明するものである。

[0178] 図 1 5 A は、本発明の一実施の形態における、検索開始ノードより配列を検索し、インデックスキーの最小値を含むリーフノードを求め、リーフノードのキーリンク表に連結したインデックスキーを順次取り出す処理の前段の処理フロー例を説明する図である。図 1 5 A に例示する前段の処理は、検索開始ノードより配列を検索し、インデックスキーの最小値を含むリーフノードを求め、リーフノードから、分類参照ポインタを取り出すものである。

[0179] 図 1 5 A に示すように、まずステップ S 1 5 0 1 において、配列番号に、検索開始ノードの配列番号を設定する。ここで検索開始ノードの配列番号は、図 7 B に示すステップ S 7 1 1 で設定された分類ツリーのルートノードの配列番号か、ステップ S 7 2 4 で設定されたノード [1] の配列番号である。

[0180] 次にステップ S 1 5 0 1 a において、探索経路スタックに配列番号を格納する。そして、ステップ S 1 5 0 2 に進み、配列から、配列番号の指す配列要素をノードとして読み出し、ステップ S 1 5 0 3 において、該読み出したノードから、ノード種別を取り出してステップ S 1 5 0 4 に進む。

[0181] ステップ S 1 5 0 4 では、ステップ S 1 5 0 3 で取り出したノード種別はブランチノードを示すものか判定する。ノード種別がブランチノードを示すものであれば、ステップ S 1 5 0 5 に進み、ステップ S 1 5 0 2 で読み出したノードから、代表ノード番号を取り出し、ステップ S 1 5 0 6 において、該取り出した代表ノード番号に値 0 を加えて、配列番号を得て、ステップ S 1 5 0 1 a に戻る。

[0182] 一方、ステップ S 1 5 0 4 において、ステップ S 1 5 0 3 で取り出したノード種別がリーフノードと判定されると、ステップ S 1 5 0 8 に進み、ステップ S 1 5 0 2 で読み出したノードから、分類参照ポインタを取り出して、図 1 5 B に示すステップ S 1 5 1 1 に進む。

[0183] 図 1 5 B は、本発明の一実施の形態における、検索開始ノードより配列を検索し、インデックスキーの最小値を含むリーフノードを求め、リーフノード

のキーリンク表に連結したインデックスキーを順次取り出す処理の後段の処理フロー例を説明する図である。図 1 5 B に例示する後段の処理は、前段の処理で求めたインデックスキーの最小値を含むリーフノードのキーリンク表に連結するインデックスキーを順次取り出して出力先に出力するものである。

- [0184] 図 1 5 B に示すように、ステップ S 1 5 1 1 において、図 1 5 A に示すステップ S 1 5 0 8 で取り出した分類参照ポインタの指すキー分類表の先頭リンクと末尾リンク、キーの出力先を読み出し、ステップ S 1 5 1 2 において、読出ポインタに、該読み出した先頭リンクを設定する。
- [0185] 次にステップ S 1 5 1 3 において、読出ポインタの指すキーリンク表のキー参照ポインタとリンクを読み出す。ここで読み出しポインタは、ステップ S 1 5 1 2 あるいは後記ステップ S 1 5 1 6 で設定されたものである。
- [0186] 次にステップ S 1 5 1 3 a において、インデックスキーの格納領域より、ステップ S 1 5 1 3 で読み出したキー参照ポインタの指すキーを読み出し、ステップ S 1 5 1 4 において、該読み出したキーをステップ S 1 5 1 1 で読み出したキーの出力先に出力し、ステップ S 1 5 1 5 に進む。
- [0187] ステップ S 1 5 1 5 では、読出ポインタと末尾リンクは一致するか判定し、一致しなければステップ S 1 5 1 6 において、読出ポインタに、ステップ S 1 5 1 3 で読み出したリンクを設定してステップ S 1 5 1 3 に戻る。
- [0188] ステップ S 1 5 1 5 において、読出ポインタと末尾リンクが一致すると判定されると、図 1 5 A に示す前段の処理で求めたインデックスキーの最小値を含むリーフノードのキーリンク表に連結するインデックスキーは全て出力先に出力されたので、処理を終了する。
- [0189] 以上、本発明の実施の形態に係るビット列キーの分類手法と分配手法を実現する処理フローについて説明をした。これらの処理フローを図 4 に例示するデータ処理装置 3 0 1 のようなコンピュータに実行させるプログラムにより、本発明に係るビット列キー分類装置とビット列キー分配装置をコンピュータ上に構築可能なことは明らかである。

そこで、本発明に係るビット列キー分類装置とビット列キー分配装置の機能構成例について、以下に説明する。

[0190] 図 1 6 は、本発明の一実施の形態におけるビット列キー分類・分配装置の機能ブロック構成例を説明する図である。

図 1 6 に示すように、ビット列キー分類装置 5 0 0 は、分類ツリー最大段数取得手段 5 1 0、分類ツリー生成手段 5 2 0、キー記憶手段 5 5 0 を含んで構成され、分類ツリー最大段数取得手段 5 1 0 で取得した最大段数の制限の中で、キー記憶手段 5 5 0 に記憶された分類対象のビット列キーを分類キーとして読み出し、分類ツリー 5 3 0 とキー位置検索表 5 4 0 を生成することにより、分類対象のビット列キーを分類する。

キー位置検索表 5 4 0 は、キー分類表 3 2 1 とキーリンク表 3 2 2 とすることができ。

[0191] ビット列キー分配装置 6 0 0 は、リーフノード取り出し手段 6 1 0 と分類キー出力手段 6 2 0 を含んで構成され、リーフノード取り出し手段 6 1 0 は分類ツリー 5 3 0 からリーフノードを順次取り出し、分類キー出力手段 6 2 0 は、リーフノード取り出し手段 6 1 0 が取り出したリーフノードからキーアクセス情報を読み出し、キーアクセス情報を用いてキー位置検索表からキーの位置情報を取り出し、キーの位置情報に基づいてキーを前記キー記憶手段から読み出して出力先に出力する。なお、キーの出力先は、分類処理の終了時にキー分類表に設定することもできるが、分類対象のキーを分類するブロックに対応して設定しておき、分類ツリーからリーフノードを取り出す毎に、決定することもできる。

請求の範囲

[請求項1]

ビット列からなる分類対象のキーを分類キーとして選択し、該分類キーの値に基づいて複数のブロックに分類するビット列キー分類装置において、

前記分類対象のキーを記憶するキー記憶手段と、

ルートノードと、隣接した記憶領域に配置されるブランチノードとリーフノードまたはブランチノード同士またはリーフノード同士のノード対、からなるツリーであって、前記ルートノードは、ツリーの始点を表すノードであって、該ツリーのノードが1つのときは前記リーフノード、ツリーのノードが2つ以上のときは前記ブランチノードであり、前記ブランチノードは、前記分類キーの弁別ビット位置とリンク先のノード対の一方のノードである代表ノードの位置を示す位置情報を含み、前記リーフノードは前記キー記憶手段に記憶されている前記分類キーの位置情報を取得するために用いるキーアクセス情報を含み、

前記ツリーの任意のノードを検索開始ノードとし、前記ブランチノードにおいて、該ブランチノードに含まれる弁別ビット位置の前記分類キーのビット値に応じてリンク先のノード対の代表ノードかあるいはそれと隣接した記憶領域に配置されたノードにリンクすることを順次前記リーフノードに至るまで繰り返すことにより、前記リーフノードに格納されたキーアクセス情報を、前記検索開始ノードをルートノードとする前記ツリーの任意の部分木の前記分類キーによる検索結果とするように構成された分類ツリーを生成する分類ツリー生成手段と、

前記分類対象のキーを分類する複数のブロックの数に応じた前記分類ツリーの最大段数を取得する分類ツリー最大段数取得手段と、

を備え、

前記分類ツリー生成手段は、

前記キー記憶手段から1つのキーを分類キーとして選択して該分類キーの位置情報を取得し、該取得した分類キーの位置情報を取得するために用いる前記キーアクセス情報を含むリーフノードをルートノードとする前記分類ツリーを生成するとともに、前記キーアクセス情報を用いて前記キーの位置情報を取得するための情報を格納するキー位置検索表を生成するルートノード生成手段と、

前記キー記憶手段からさらに順次分類対象のキーを分類キーとして選択して該分類キーの位置情報を取得し、前記分類ツリー最大段数取得手段で取得した分類ツリーの最大段数の範囲で、該取得した分類キーの位置情報を取得するために用いる前記キーアクセス情報を含むリーフノードを前記分類ツリーに挿入するとともに、前記キーアクセス情報を用いて前記分類キーの位置情報を取得するための情報を格納するキー位置検索表を生成するか、あるいは、既存のリーフノードに含まれるキーアクセス情報を用いて分類キーの位置情報を取得する情報を格納するキー位置検索表に、該既存のリーフノードに含まれるキーアクセス情報を用いて前記分類キーとして選択された分類対象のキーの位置情報を取得する情報を追加する分類キー挿入手段と、を含み、前記分類対象のキーの全てを前記分類キーとして選択して前記分類ツリーを生成するとともに、該分類ツリーの各リーフノードに含まれるキーアクセス情報を用いて前記分類キーの位置情報を取得する情報を格納する前記キー位置検索表を生成することにより、前記分類対象のキーの全てを、前記各リーフノードに対応する複数のブロックに分類することを特徴とするビット列キー分類装置。

[請求項2]

請求項1記載のビット列キー分類装置において、

前記キー位置検索表は、前記複数のブロックに対応するエントリを有するキー分類表と、前記分類対象のキーに対応するエントリを有するキーリンク表から構成され、

前記キー分類表のエントリは、前記キーリンク表のエントリを指す

キー管理ポインタが格納される領域である先頭リンクと末尾リンクを含み、

前記キーリンク表のエントリは、前記対応する分類対象のキーが記憶された前記キー記憶手段の位置情報を指すキー参照ポインタを格納する領域と、前記キー管理ポインタを格納する領域であるリンクを含むものであり、

前記キーアクセス情報は、前記キー分類表のエントリを指す分類参照ポインタであることを特徴とするビット列キー分類装置。

[請求項3]

請求項2記載のビット列キー分類装置において、

前記キー分類表のエントリは、さらに、前記対応するブロックに分類されているキーの最小値を格納する領域である最小値キーと前記対応するブロックに分類されているキーの最大値を格納する領域である最大値キーを含み、

前記ルートノード生成手段は、前記キー記憶手段から1つのキーを分類キーとして選択して読出し、該分類キーの位置情報を取得するとともに、前記キー分類表の空エントリを指す分類参照ポインタを取得し、該分類参照ポインタ含むリーフノードをルートノードとする前記分類ツリーを生成し、さらに、前記キーリンク表の空エントリを指すキー管理ポインタを取得し、該キー管理ポインタを前記取得したキー分類表のエントリの前記先頭リンクと前記末尾リンクに格納し、前記最小値キーと最大値キーに前記読み出した分類キーを格納するとともに、前記取得した位置情報を前記取得したキーリンク表のエントリのキー参照ポインタに格納するものであり、

前記分類キー挿入手段は、前記キー記憶手段からさらに順次分類対象のキーを分類キーとして選択して読み出し、該分類キーの位置情報を取得するとともに、前記ルートノードを検索開始ノードとして前記分類キーによる検索結果である分類参照ポインタを求める検索手段と、前記キーリンク表の空エントリを指すキー管理ポインタを取得し、該

キー管理ポインタの指すキーリンク表のキー参照ポインタに前記検索手段が取得した位置情報を格納するとともに、前記分類参照ポインタの指す前記キー分類表のエントリの末尾リンクの指すキーリンク表のリンクに、前記キー管理ポインタを格納することにより、前記分類参照ポインタを用いて分類キーの位置情報を取得するための情報を前記キー分類表とキーリンク表に追加する第1の分類キー連結手段と

前記キーリンク表の空エントリを指すキー管理ポインタを取得し、該キー管理ポインタの指すキーリンク表のキー参照ポインタに前記検索手段が取得した位置情報を格納するとともに、前記分類参照ポインタの指す前記キー分類表のエントリの末尾リンクの指すキーリンク表のリンクに、前記キー管理ポインタを格納することにより、前記分類参照ポインタを用いて分類キーの位置情報を取得するための情報を前記キー分類表とキーリンク表に追加するとともに、前記分類キーが該分類参照ポインタの指すキー分類表の最小値キーより小さければ、該最小値キーに該分類キーを書き込み、一方、前記分類キーが前記分類参照ポインタの指すキー分類表の最大値キーより大きければ、該最大値キーに該分類キーを書き込む第2の分類キー連結手段と、

前記検索手段が求めた分類参照ポインタの指す前記キー分類表のエントリの最小値キーあるいは最大値キーの一方を検索結果キーとし、該検索結果キーと前記分類キーの間でビット列比較を行い、該ビット列比較で異なるビット値となる先頭のビット位置である差分ビット位置を求め、前記検索手段が前記分類参照ポインタを求めたときのリンク経路のブランチノードの弁別ビット位置と前記差分ビット位置との相対的位置関係により、前記分類キーの位置情報を取得するために用いる分類参照ポインタを含むリーフノードと該リーフノードと対をなすノードからなるノード対を前記分類ツリーに挿入したときに該ノード対のリンク元となるノードの位置情報を該ノード対の挿入位置として決定するノード対挿入位置決定手段と、

前記分類キーの位置情報を取得するために用いる分類参照ポインタを含むリーフノードと対をなすノードの内容を、前記挿入位置のノードの内容とし、該挿入位置のノードのノード種別にはブランチノードを示すものを書き込み、弁別ビット位置には前記差分ビット位置を書き込み、代表ノード番号には、前記ノード対挿入位置決定手段で挿入位置を求めたノード対の代表ノードの位置情報を書き込むことで該ノード対を挿入するノード対挿入手段と、

前記ノード対挿入位置決定手段が決定したノード対の挿入位置に該ノード対を挿入したときに、該挿入位置のノードより下位のリーフノードの段数が前記最大段数を越さないかチェックを行い、前記最大段数を超すリーフノードがあれば、そのリーフノードを含むノード対の直近上位のノードである親ノードをリーフノードとし、前記ノード対を構成するリーフノードそれぞれの分類参照ポインタと、該分類参照ポインタの指す前記キー分類表のデータに基づいて、前記リーフノードとした親ノードの分類参照ポインタを決定するとともに、該分類参照ポインタの指すキー分類表のデータとキーリンク表のデータを更新し、前記ノード対を削除することを、前記挿入位置のノードより下位の全てのリーフノードに対して行うことにより、ノード対を挿入しても前記最大段数を超えないことを保証する処理を行うノード対連結手段と、を含み、

前記読み出された分類キーの値が、前記検索手段が求めた分類参照ポインタの指す前記キー分類表のエントリの最小値キーの値と最大値キーの値の間の範囲のものである場合に、

前記第 1 の分類キー連結手段が、前記分類参照ポインタを用いて分類キーの位置情報を取得するための情報を前記キー分類表とキーリンク表に追加し、

前記読み出された分類キーの値が、前記検索手段が求めた分類参照ポインタの指す前記キー分類表のエントリの最小値キーの値と最大値キ

一の値の間の範囲外のものである場合に、
前記ノード対挿入位置決定手段が、前記ノード対のリンク元となるノードの位置情報を該ノード対の挿入位置として決定し、
該決定された挿入位置のノードがリーフノードであって、該リーフノードの段数が前記最大段数のとき、前記第2の分類キー連結手段が、前記分類参照ポインタを用いて分類キーの位置情報を取得するための情報を前記キー分類表とキーリンク表に追加するとともに、前記分類キーが該分類参照ポインタの指すキー分類表の最小値キーより小さければ、該最小値キーに該分類キーを書き込み、一方、前記分類キーが前記分類参照ポインタの指すキー分類表の最大値キーより大きければ、該最大値キーに該分類キーを書き込み、
前記決定された挿入位置のノードがリーフノードであって、該リーフノードの段数が前記最大段数ではないとき、前記ノード対挿入手段が前記ノード対を挿入し、
前記ノード対挿入位置決定手段が決定した挿入位置のノードがブランチノードのとき、前記ノード対連結手段が前記ノード対を挿入しても前記最大段数を超えないことを保証する処理を行い、前記ノード対挿入手段が前記ノード対を挿入する、
ことを特徴とするビット列キー分類装置。

[請求項4] 請求項3記載のビット列キー分類装置において、前記分類ツリーは配列に配置され、前記代表ノードの位置を示す位置情報は、該代表ノードが配置された前記配列の配列要素の配列番号であることを特徴とするビット列キー分類装置。

[請求項5] 請求項1記載のビット列キー分類装置により複数のブロックに分類されたキーをブロック毎に取り出して出力するビット列キー分配装置において、

前記分類ツリーから順次リーフノードを取り出すリーフノード取り出し手段と、

該取り出したリーフノードから、前記キーアクセス情報を読み出し、該キーアクセス情報を用いて前記キー位置検索表からキーの位置情報を取り出し、該キーの位置情報に基づいて該キーを前記キー記憶手段から読み出して出力先に出力する分類キー出力手段と、

を備えることを特徴とするビット列キー分配装置。

[請求項6]

請求項5記載のビット列キー分配装置において、

前記キー位置検索表は、前記複数のブロックに対応するエントリを有するキー分類表と、前記分類対象のキーに対応するエントリを有するキーリンク表から構成され、

前記キー分類表のエントリは、前記キーリンク表のエントリを指すキー管理ポインタが格納される領域である先頭リンクと末尾リンクを含み、

前記キーリンク表のエントリは、前記対応する分類対象のキーが記憶された前記キー記憶手段の位置情報を指すキー参照ポインタを格納する領域と、前記キー管理ポインタを格納する領域であるリンクを含むものであり、

前記キーアクセス情報は、前記キー分類表のエントリを指す分類参照ポインタであることを特徴とするビット列キー分配装置。

[請求項7]

請求項6記載のビット列キー分配装置において、前記分類ツリーは配列に配置され、前記代表ノードの位置を示す位置情報は、該代表ノードが配置された前記配列の配列要素の配列番号であることを特徴とするビット列キー分配装置。

[請求項8]

分類対象のキーを記憶するキー記憶手段を備えたビット列分類装置が、ビット列からなる分類対象のキーを分類キーとして選択し、該分類キーの値に基づいて複数のブロックに分類するビット列キー分類方法において、

前記分類対象のキーを分類する複数のブロックの数に応じた後記分類ツリーの最大段数を取得する分類ツリー最大段数取得ステップと、

ルートノードと、隣接した記憶領域に配置されるブランチノードとリーフノードまたはブランチノード同士またはリーフノード同士のノード対、からなるツリーであって、前記ルートノードは、ツリーの始点を表すノードであって、該ツリーのノードが1つのときは前記リーフノード、ツリーのノードが2つ以上のときは前記ブランチノードであり、前記ブランチノードは、前記分類キーの弁別ビット位置とリンク先のノード対の一方のノードである代表ノードの位置を示す位置情報を含み、前記リーフノードは前記キー記憶手段に記憶されている前記分類キーの位置情報を取得するために用いるキーアクセス情報を含み、

前記ツリーの任意のノードを検索開始ノードとし、前記ブランチノードにおいて、該ブランチノードに含まれる弁別ビット位置の前記分類キーのビット値に応じてリンク先のノード対の代表ノードかあるいはそれと隣接した記憶領域に配置されたノードにリンクすることを順次前記リーフノードに至るまで繰り返すことにより、前記リーフノードに格納されたキーアクセス情報を、前記検索開始ノードをルートノードとする前記ツリーの任意の部分木の前記分類キーによる検索結果とするように構成された分類ツリーを生成する分類ツリー生成ステップと、

を備え、

前記分類ツリー生成ステップは、

前記キー記憶手段から1つのキーを分類キーとして選択して該分類キーの位置情報を取得し、該取得した分類キーの位置情報を取得するために用いる前記キーアクセス情報を含むリーフノードをルートノードとする前記分類ツリーを生成するとともに、前記キーアクセス情報を用いて前記キーの位置情報を取得するための情報を格納するキー位置検索表を生成するルートノード生成ステップと、

前記キー記憶手段からさらに順次分類対象のキーを分類キーとして選

択して該分類キーの位置情報を取得し、前記分類ツリー最大段数取得手段で取得した分類ツリーの最大段数の範囲で、該取得した分類キーの位置情報を取得するために用いる前記キーアクセス情報を含むリーフノードを前記分類ツリーに挿入するとともに、前記キーアクセス情報を用いて前記分類キーの位置情報を取得するための情報を格納するキー位置検索表を生成するか、あるいは、既存のリーフノードに含まれるキーアクセス情報を用いて分類キーの位置情報を取得する情報を格納するキー位置検索表に、該既存のリーフノードに含まれるキーアクセス情報を用いて前記分類キーとして選択された分類対象のキーの位置情報を取得する情報を追加する分類キー挿入ステップと、を含み、

前記分類対象のキーの全てを前記分類キーとして選択して前記分類ツリーを生成するとともに、該分類ツリーの各リーフノードに含まれるキーアクセス情報を用いて前記分類キーの位置情報を取得する情報を格納する前記キー位置検索表を生成することにより、前記分類対象のキーの全てを、前記各リーフノードに対応する複数のブロックに分類することを特徴とするビット列キー分類方法。

[請求項9]

請求項8記載のビット列キー分類方法において、

前記キー位置検索表は、前記複数のブロックに対応するエントリを有するキー分類表と、前記分類対象のキーに対応するエントリを有するキーリンク表から構成され、

前記キー分類表のエントリは、前記キーリンク表のエントリを指すキー管理ポインタが格納される領域である先頭リンクと末尾リンクを含み、

前記キーリンク表のエントリは、前記対応する分類対象のキーが記憶された前記キー記憶手段の位置情報を指すキー参照ポインタを格納する領域と、前記キー管理ポインタを格納する領域であるリンクを含むものであり、

前記キーアクセス情報は、前記キー分類表のエントリを指す分類参照ポインタであることを特徴とするビット列キー分類方法。

[請求項10]

請求項9記載のビット列キー分類方法において、

前記キー分類表のエントリは、さらに、前記対応するブロックに分類されているキーの最小値を格納する領域である最小値キーと前記対応するブロックに分類されているキーの最大値を格納する領域である最大値キーを含み、

前記ルートノード生成ステップは、前記キー記憶手段から1つのキーを分類キーとして選択して読出し、該分類キーの位置情報を取得するとともに、前記キー分類表の空エントリを指す分類参照ポインタを取得し、該分類参照ポインタ含むリーフノードをルートノードとする前記分類ツリーを生成し、さらに、前記キーリンク表の空エントリを指すキー管理ポインタを取得し、該キー管理ポインタを前記取得したキー分類表のエントリの前記先頭リンクと前記末尾リンクに格納し、前記最小値キーと最大値キーに前記読み出した分類キーを格納するとともに、前記取得した位置情報を前記取得したキーリンク表のエントリのキー参照ポインタに格納するものであり、

前記分類キー挿入ステップは、前記キー記憶手段からさらに順次分類対象のキーを分類キーとして選択して読み出し、該分類キーの位置情報を取得するとともに、前記ルートノードを検索開始ノードとして前記分類キーによる検索結果である分類参照ポインタを求める検索ステップと、

前記キーリンク表の空エントリを指すキー管理ポインタを取得し、該キー管理ポインタの指すキーリンク表のキー参照ポインタに前記検索ステップにおいて取得した位置情報を格納するとともに、前記分類参照ポインタの指す前記キー分類表のエントリの末尾リンクの指すキーリンク表のリンクに、前記キー管理ポインタを格納することにより、前記分類参照ポインタを用いて分類キーの位置情報を取得するための

情報を前記キー分類表とキーリンク表に追加する第1の分類キー連結ステップと

前記キーリンク表の空エントリを指すキー管理ポインタを取得し、該キー管理ポインタの指すキーリンク表のキー参照ポインタに前記検索ステップにおいて取得した位置情報を格納するとともに、前記分類参照ポインタの指す前記キー分類表のエントリの末尾リンクの指すキーリンク表のリンクに、前記キー管理ポインタを格納することにより、前記分類参照ポインタを用いて分類キーの位置情報を取得するための情報を前記キー分類表とキーリンク表に追加するとともに、前記分類キーが該分類参照ポインタの指すキー分類表の最小値キーより小さければ、該最小値キーに該分類キーを書き込み、一方、前記分類キーが前記分類参照ポインタの指すキー分類表の最大値キーより大きければ、該最大値キーに該分類キーを書き込む第2の分類キー連結ステップと、

前記検索手段ステップにおいて求めた分類参照ポインタの指す前記キー分類表のエントリの最小値キーあるいは最大値キーの一方を検索結果キーとし、該検索結果キーと前記分類キーの間でビット列比較を行い、該ビット列比較で異なるビット値となる先頭のビット位置である差分ビット位置を求め、前記検索ステップにおいて前記分類参照ポインタを求めたときのリンク経路のブランチノードの弁別ビット位置と前記差分ビット位置との相対的位置関係により、前記分類キーの位置情報を取得するために用いる分類参照ポインタを含むリーフノードと該リーフノードと対をなすノードからなるノード対を前記分類ツリーに挿入したときに該ノード対のリンク元となるノードの位置情報を該ノード対の挿入位置として決定するノード対挿入位置決定ステップと、

前記分類キーの位置情報を取得するために用いる分類参照ポインタを含むリーフノードと対をなすノードの内容を、前記挿入位置のノード

の内容とし、該挿入位置のノードのノード種別にはブランチノードを示すものを書き込み、弁別ビット位置には前記差分ビット位置を書き込み、代表ノード番号には、前記ノード対挿入位置決定ステップにおいて挿入位置を求めたノード対の代表ノードの位置情報を書き込むことで該ノード対を挿入するノード対挿入ステップと、

前記ノード対挿入位置決定ステップにおいて決定したノード対の挿入位置に該ノード対を挿入したときに、該挿入位置のノードより下位のリーフノードの段数が前記最大段数を越さないかチェックを行い、前記最大段数を超すリーフノードがあれば、そのリーフノードを含むノード対の直近上位のノードである親ノードをリーフノードとし、前記ノード対を構成するリーフノードそれぞれの分類参照ポインタと、該分類参照ポインタの指す前記キー分類表のデータに基づいて、前記リーフノードとした親ノードの分類参照ポインタを決定するとともに、該分類参照ポインタの指すキー分類表のデータとキーリンク表のデータを更新し、前記ノード対を削除することを、前記挿入位置のノードより下位の全てのリーフノードに対して行うことにより、ノード対を挿入しても前記最大段数を超えないことを保証する処理を行うノード対連結ステップと、を含み、

前記読み出された分類キーの値が、前記検索手段が求めた分類参照ポインタの指す前記キー分類表のエントリの最小値キーの値と最大値キーの値の間の範囲のものである場合に、

前記第 1 の分類キー連結ステップにおいて、前記分類参照ポインタを用いて分類キーの位置情報を取得するための情報を前記キー分類表とキーリンク表に追加し、

前記読み出された分類キーの値が、前記検索手段が求めた分類参照ポインタの指す前記キー分類表のエントリの最小値キーの値と最大値キーの値の間の範囲外のものである場合に、

前記ノード対挿入位置決定ステップにおいて、前記ノード対のリンク

元となるノードの位置情報を該ノード対の挿入位置として決定し、
該決定された挿入位置のノードがリーフノードであって、該リーフノードの段数が前記最大段数のとき、前記第2の分類キー連結ステップにおいて、前記分類参照ポインタを用いて分類キーの位置情報を取得するための情報を前記キー分類表とキーリンク表に追加するとともに、前記分類キーが該分類参照ポインタの指すキー分類表の最小値キーより小さければ、該最小値キーに該分類キーを書き込み、一方、前記分類キーが前記分類参照ポインタの指すキー分類表の最大値キーより大きければ、該最大値キーに該分類キーを書き込み、
前記決定された挿入位置のノードがリーフノードであって、該リーフノードの段数が前記最大段数ではないとき、前記ノード対挿入手段が前記ノード対を挿入し、
前記ノード対挿入位置決定ステップにおいて決定した挿入位置のノードがブランチノードのとき、前記ノード対連結ステップにおいて前記ノード対を挿入しても前記最大段数を超えないことを保証する処理を行い、前記ノード対挿入ステップにおいて前記ノード対を挿入する、ことを特徴とするビット列キー分類方法。

[請求項11] 請求項10記載のビット列キー分類方法において、前記分類ツリーは配列に配置され、前記代表ノードの位置を示す位置情報は、該代表ノードが配置された前記配列の配列要素の配列番号であることを特徴とするビット列キー分類方法。

[請求項12] ビット列キー分配装置が、請求項1記載のビット列キー分類装置により複数のブロックに分類されたキーをブロック毎に取り出して出力するビット列キー分配方法において、

前記分類ツリーから順次リーフノードを取り出すリーフノード取り出しステップと、

該取り出したリーフノードから、前記キーアクセス情報を読み出し、該キーアクセス情報を用いて前記キー位置検索表からキーの位置情

報を取り出し、該キーの位置情報に基づいて該キーを前記キー記憶手段から読み出して出力先に出力する分類キー出力ステップと、

を備えることを特徴とするビット列キー分配方法。

[請求項13]

請求項12記載のビット列キー分配方法において、

前記キー位置検索表は、前記複数のブロックに対応するエントリを有するキー分類表と、前記分類対象のキーに対応するエントリを有するキーリンク表から構成され、

前記キー分類表のエントリは、前記キーリンク表のエントリを指すキー管理ポインタが格納される領域である先頭リンクと末尾リンクを含み、

前記キーリンク表のエントリは、前記対応する分類対象のキーが記憶された前記キー記憶手段の位置情報を指すキー参照ポインタを格納する領域と、前記キー管理ポインタを格納する領域であるリンクを含むものであり、

前記キーアクセス情報は、前記キー分類表のエントリを指す分類参照ポインタであることを特徴とするビット列キー分配方法。

[請求項14]

請求項13記載のビット列キー分配方法において、前記分類ツリーは配列に配置され、前記代表ノードの位置を示す位置情報は、該代表ノードが配置された前記配列の配列要素の配列番号であることを特徴とするビット列キー分配方法。

[請求項15]

請求項8～請求項11のいずれか1項記載のビット列キー分類方法をコンピュータに実行させることを特徴とするプログラム。

[請求項16]

請求項8～請求項11のいずれか1項記載のビット列キー分類方法をコンピュータに実行させるプログラムを記憶したことを特徴とするコンピュータ読み取り可能な記憶媒体。

[請求項17]

請求項12～請求項14のいずれか1項記載のビット列キー分配方法をコンピュータに実行させることを特徴とするプログラム。

[請求項18]

請求項12～請求項14のいずれか1項記載のビット列キー分配方

法をコンピュータに実行させるプログラムを記憶したことを特徴とするコンピュータ読み取り可能な記憶媒体。

[請求項19]

キー記憶手段に記憶され、複数のブロックに分類されたビット列キーを出力先に出力するビット列キー分配装置に用いられるデータ構造において、

ルートノードと、隣接した記憶領域に配置されるブランチノードとリーフノードまたはブランチノード同士またはリーフノード同士のノード対、からなるツリーであって、前記ルートノードは、ツリーの始点を表すノードであって、該ツリーのノードが1つのときは前記リーフノード、ツリーのノードが2つ以上のときは前記ブランチノードであり、前記ブランチノードは、前記分類キーの弁別ビット位置とリンク先のノード対の一方のノードである代表ノードの位置を示す位置情報を含み、前記リーフノードは前記キー記憶手段に記憶されている前記分類キーの位置情報を取得するために用いるキーアクセス情報を含み、

前記ツリーの任意のノードを検索開始ノードとし、前記ブランチノードにおいて、該ブランチノードに含まれる弁別ビット位置の前記分類キーのビット値に応じてリンク先のノード対の代表ノードかあるいはそれと隣接した記憶領域に配置されたノードにリンクすることを順次前記リーフノードに至るまで繰り返すことにより、前記リーフノードに格納されたキーアクセス情報を、前記検索開始ノードをルートノードとする前記ツリーの任意の部分木の前記分類キーによる検索結果とするように構成された分類ツリーと、

前記キーアクセス情報を用いて前記キーの位置情報を取得するための情報を格納するキー位置検索表と、

を備え、

前記ビット列キー分配装置が、前記分類ツリーと前記キー位置検索表により、請求項12記載のビット列キー分配方法を実行し、複数のブ

ロックに分類されたビット列キーの出力先への出力を可能とすることを特徴とするデータ構造。

[請求項20]

請求項 19 記載のデータ構造において、

前記キー位置検索表は、前記複数のブロックに対応するエントリを有するキー分類表と、前記分類対象のキーに対応するエントリを有するキーリンク表から構成され、前記キー分類表のエントリは、前記キーリンク表のエントリを指すキー管理ポインタが格納される領域である先頭リンクと末尾リンクを含み、前記キーリンク表のエントリは、前記対応する分類対象のキーが記憶された前記キー記憶手段の位置情報を指すキー参照ポインタを格納する領域と、前記キー管理ポインタを格納する領域であるリンクを含み、前記分類ツリーのリーフノードに含まれる前記キーアクセス情報は、前記キー分類表のエントリを指す分類参照ポインタであり、前記ビット列キー分配装置が、前記分類ツリー、前記キー分類表及びキーリンク表により、請求項 13 記載のビット列キー分配方法を実行し、複数のブロックに分類されたビット列キーの出力先への出力を可能とすることを特徴とするデータ構造。

[請求項21]

請求項 20 記載のデータ構造において、

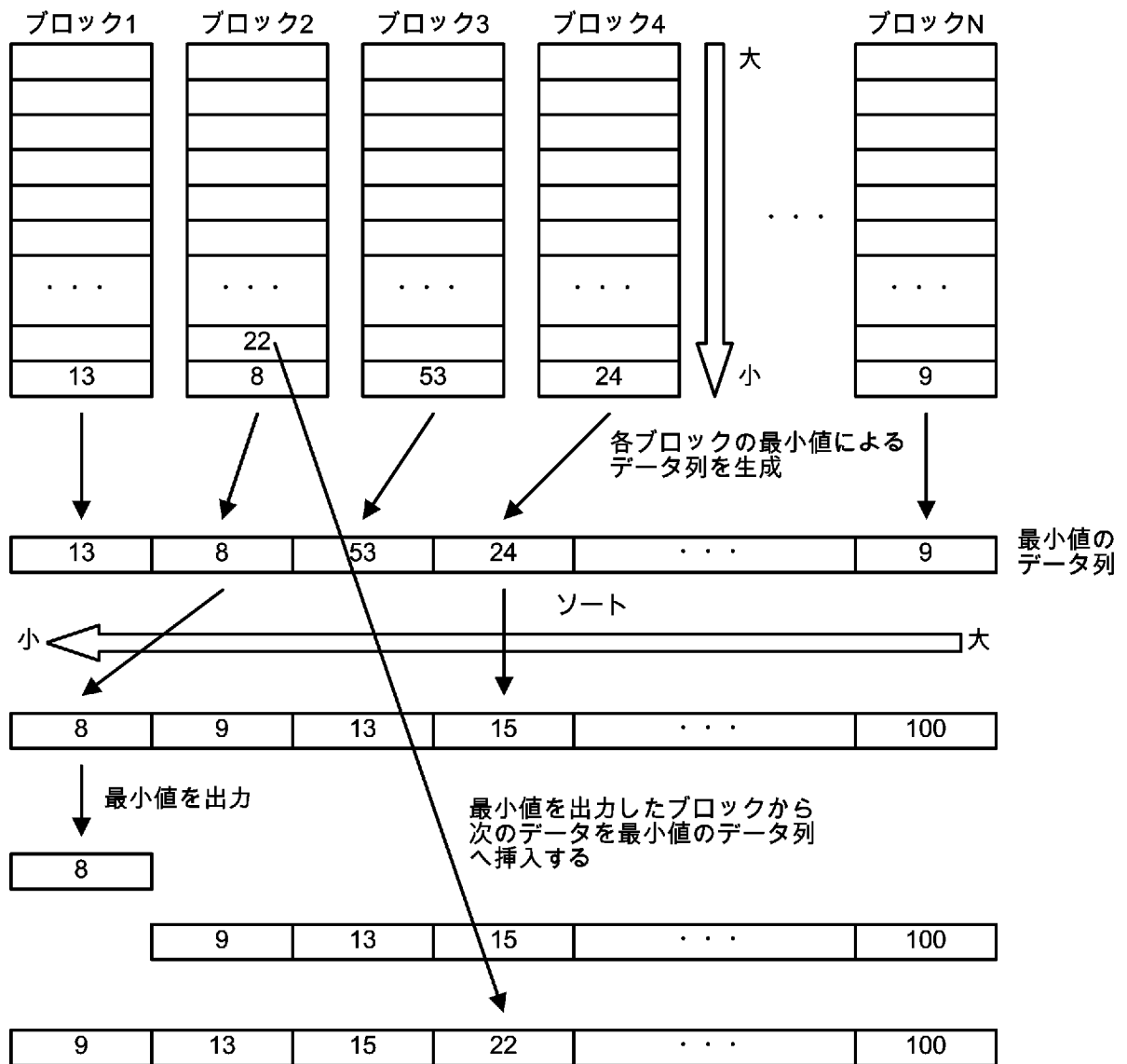
前記分類ツリーは配列に配置され、前記代表ノードの位置を示す位置情報は、該代表ノードが配置された前記配列の配列要素の配列番号であり、

前記ビット列キー分配装置が、前記分類ツリー、前記キー分類表及びキーリンク表により、請求項 14 記載のビット列キー分配方法を実行し、複数のブロックに分類されたビット列キーの出力先への出力を可能とすることを特徴とするデータ構造。

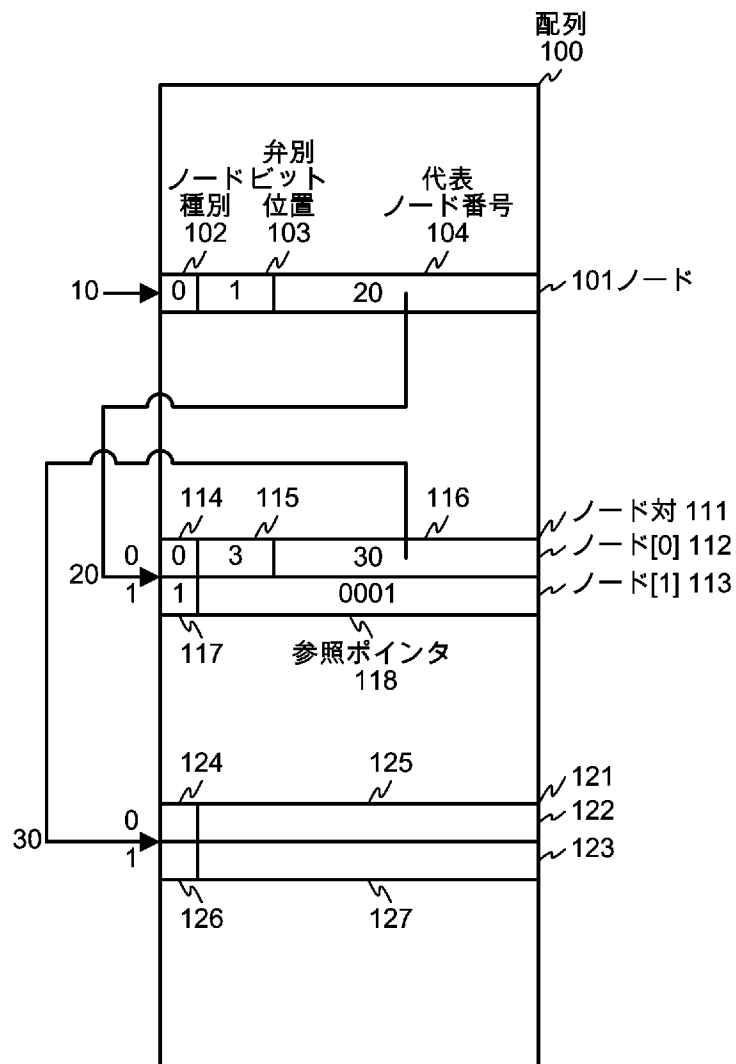
[請求項22]

請求項 19～請求項 21 のいずれか 1 項記載のデータ構造を記憶したことを特徴とするコンピュータ読み取り可能な記憶媒体。

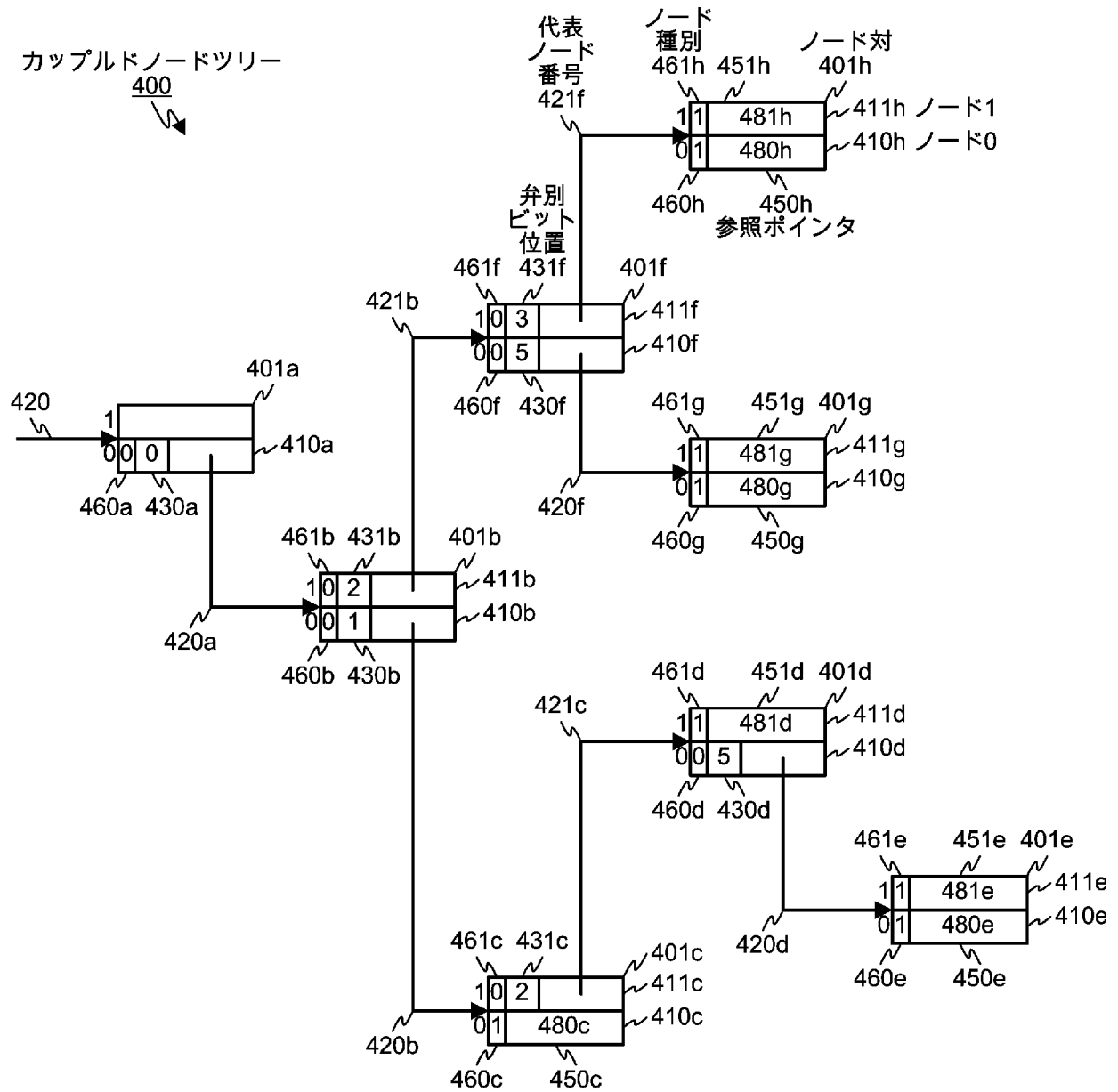
[図1]



[図2A]



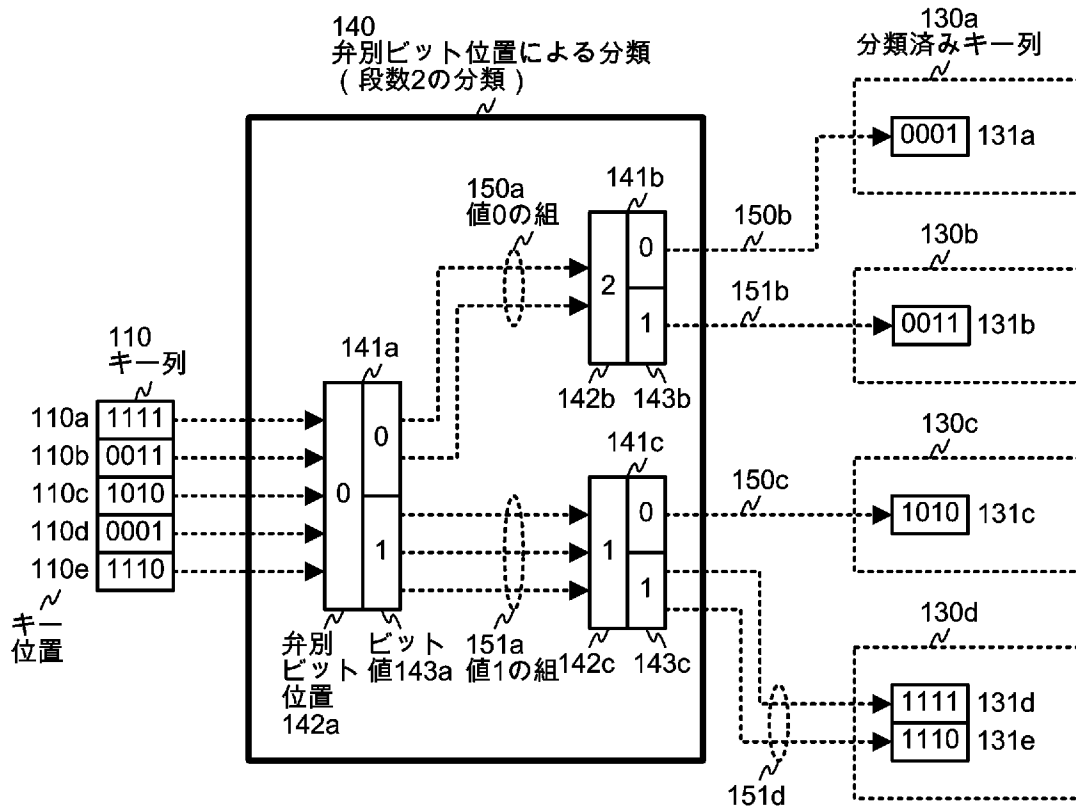
[図2B]



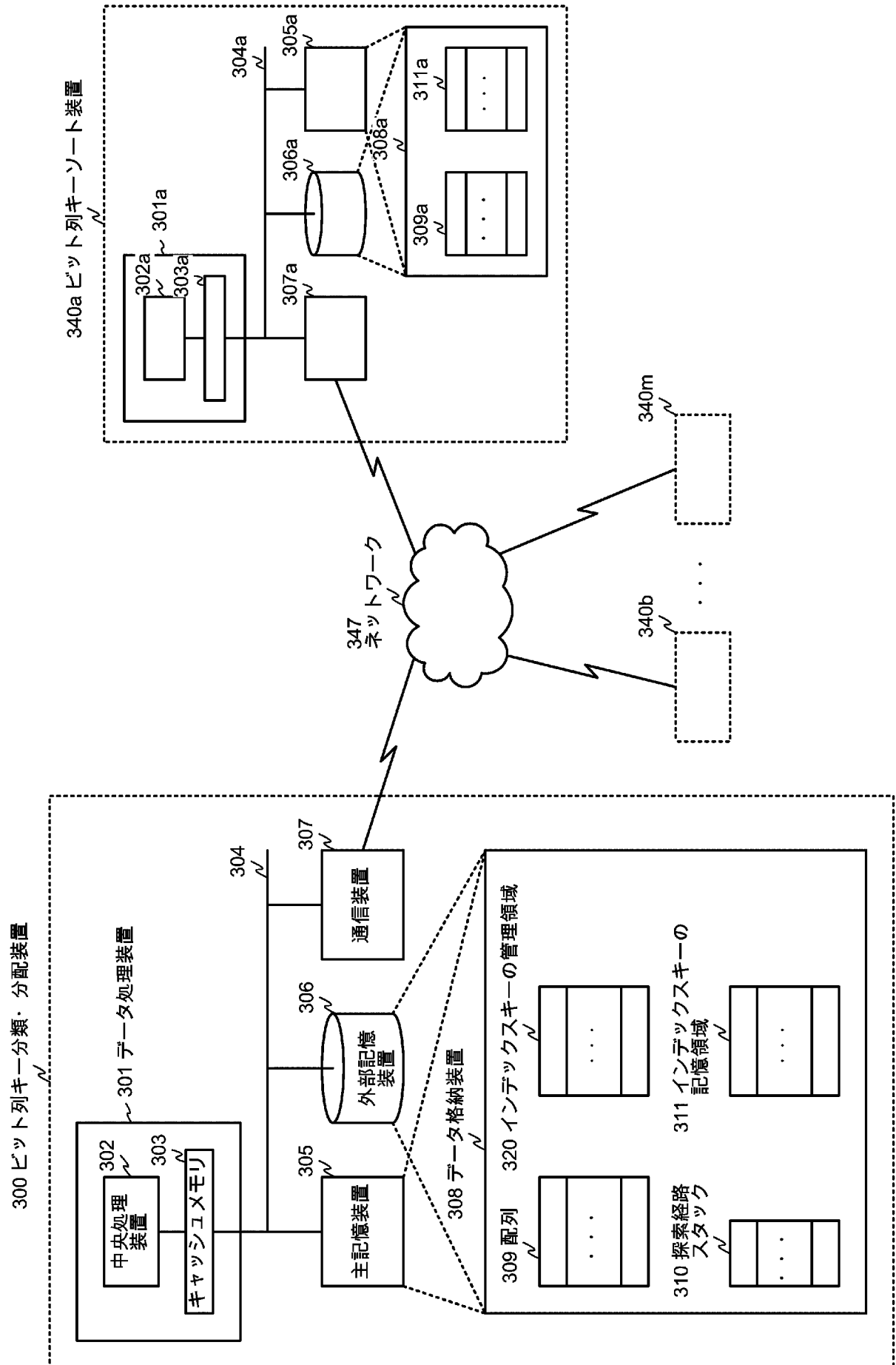
311
インデックスキーの記憶領域

参照ポインタ	481h	101100	271h インデックスキー
	480h	101011	270h
	481g	100011	271g
	480g	100010	270g
	481d	011010	271d
	481e	010011	271e
	480e	010010	270e
	480c	000111	270c

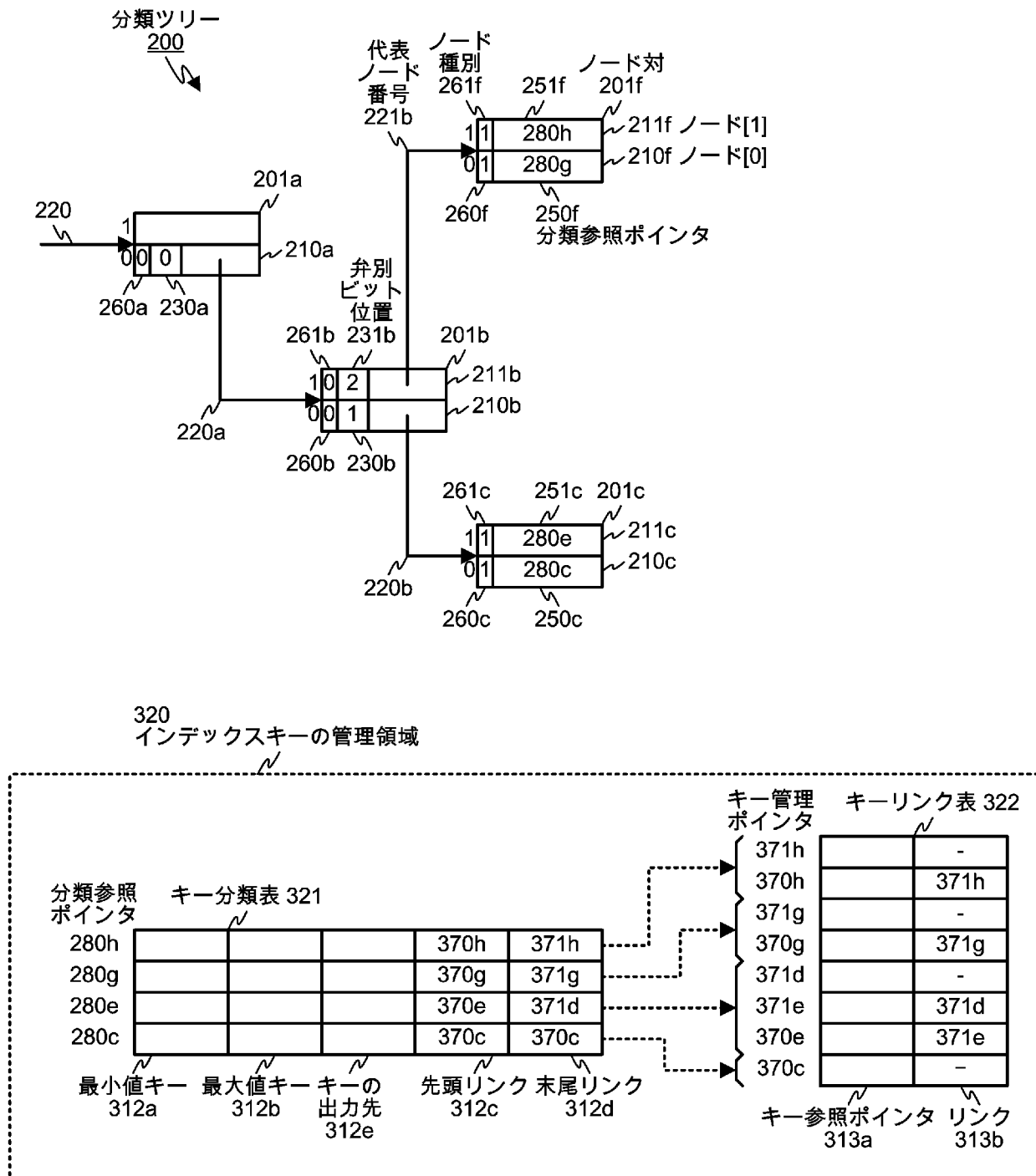
[図3]



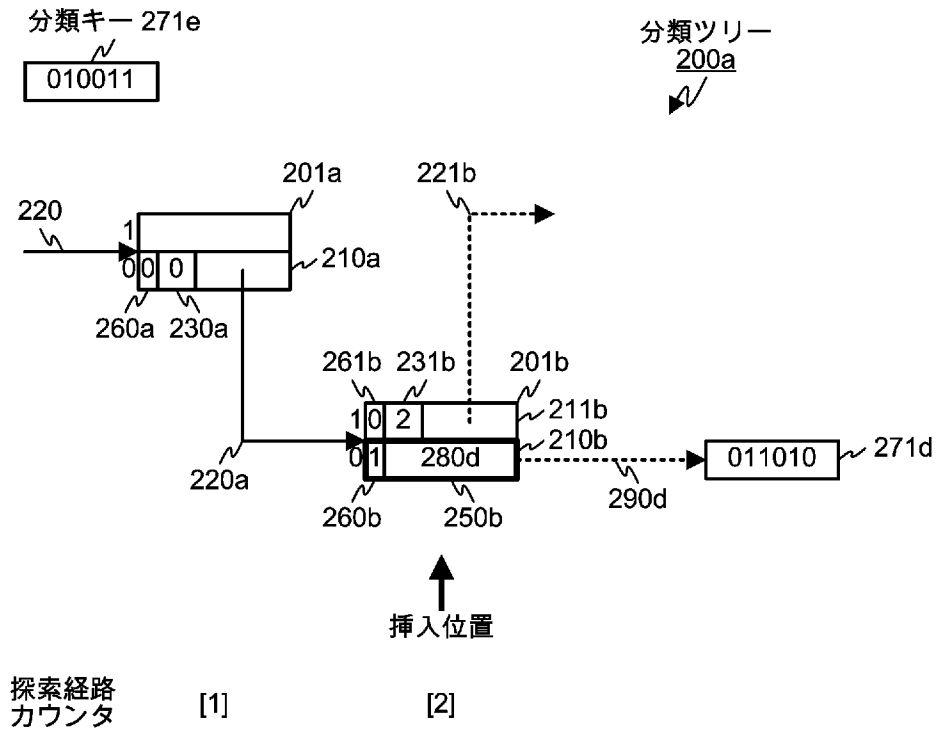
[図4]



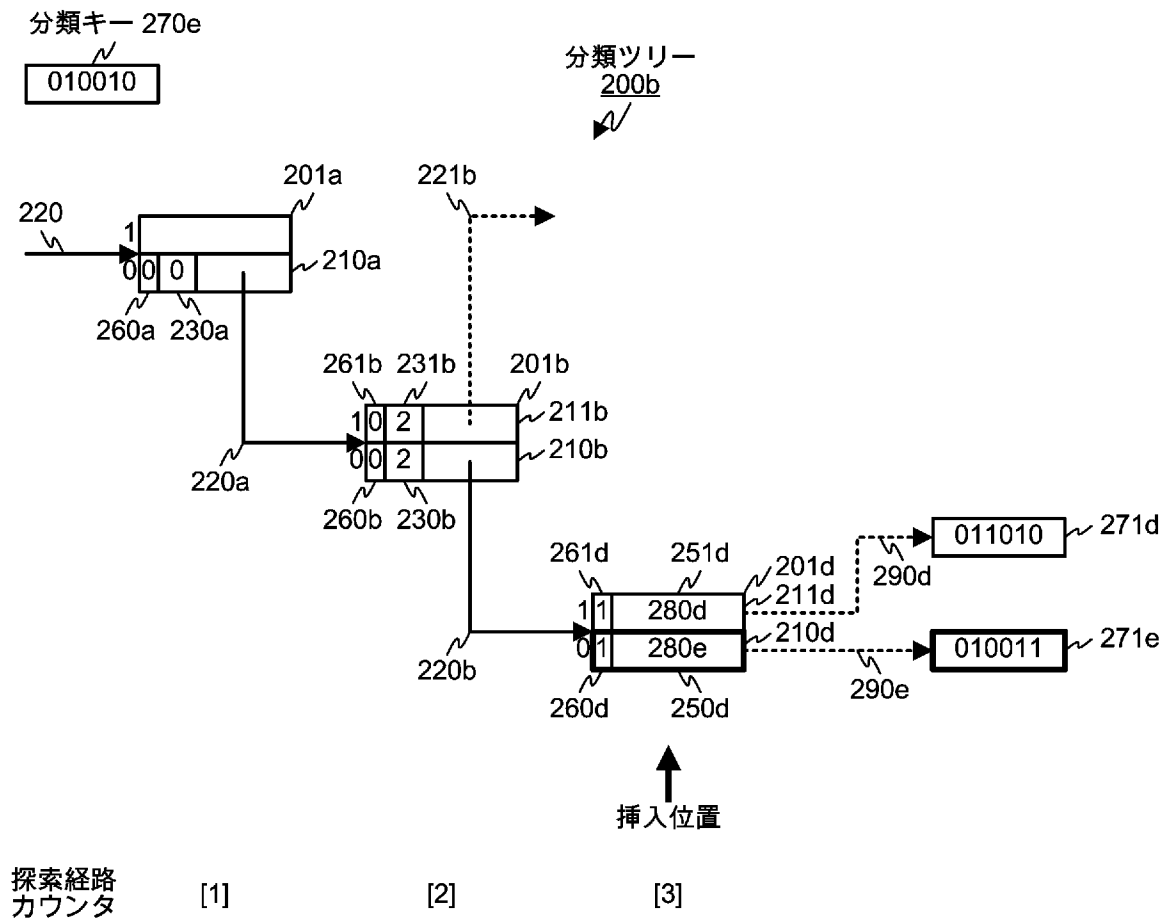
[図5]



[図6A]



[図6B]



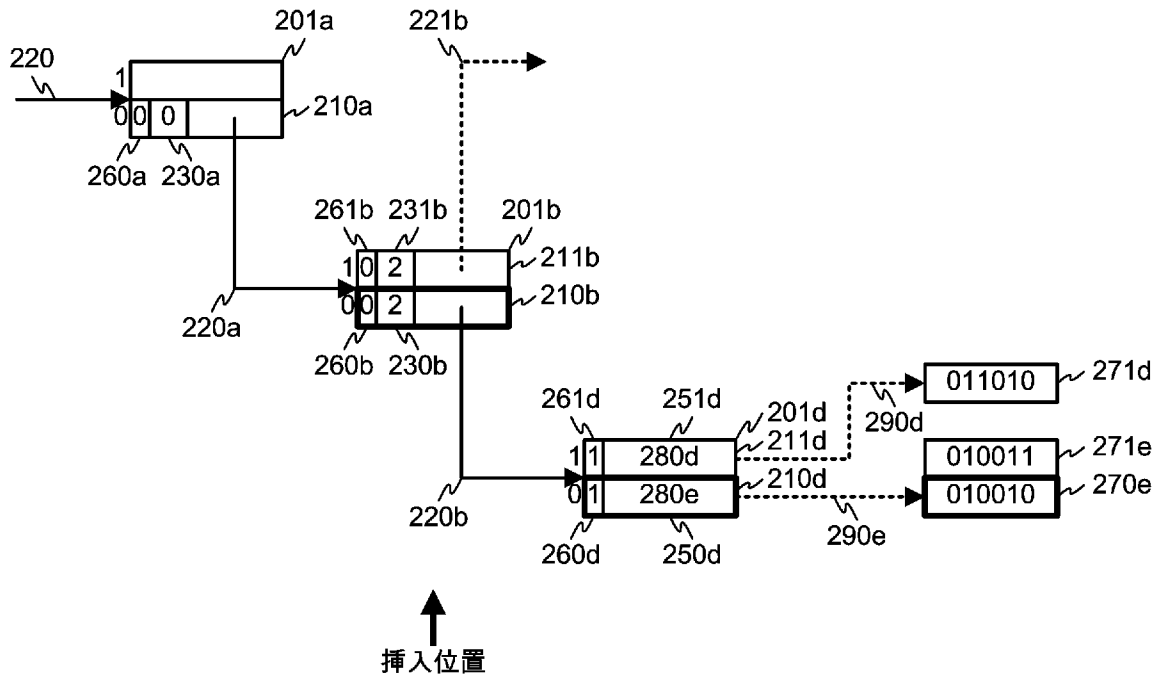
[図6C]

分類キー 270c

000111

分類ツリー

200c



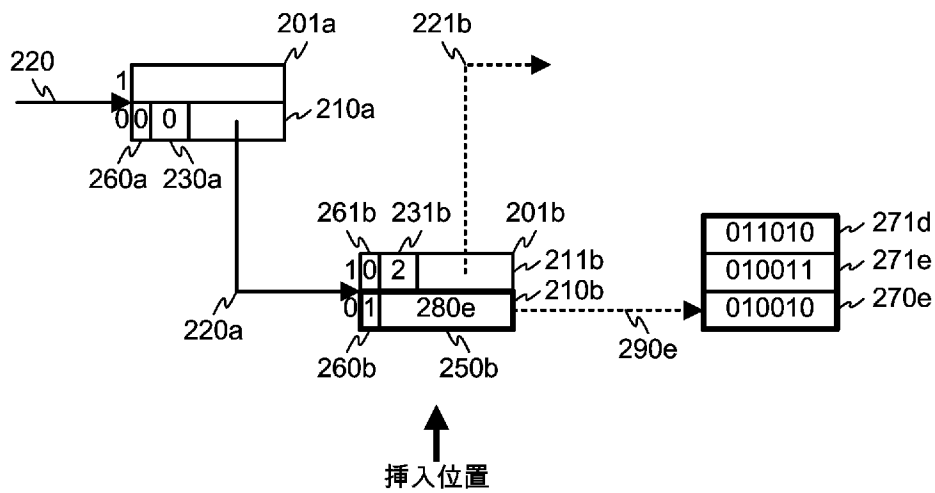
[図6D]

分類キー 270c

000111

分類ツリー

200d

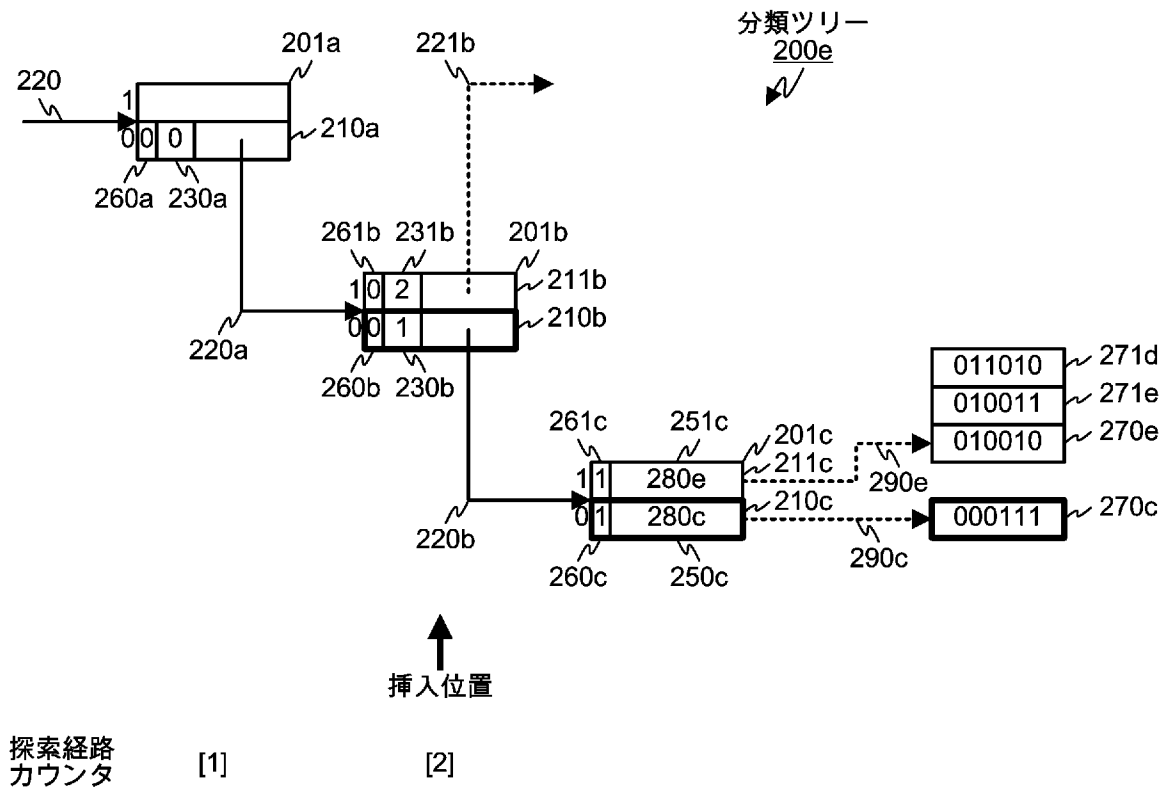
探索経路
カウンタ

[1]

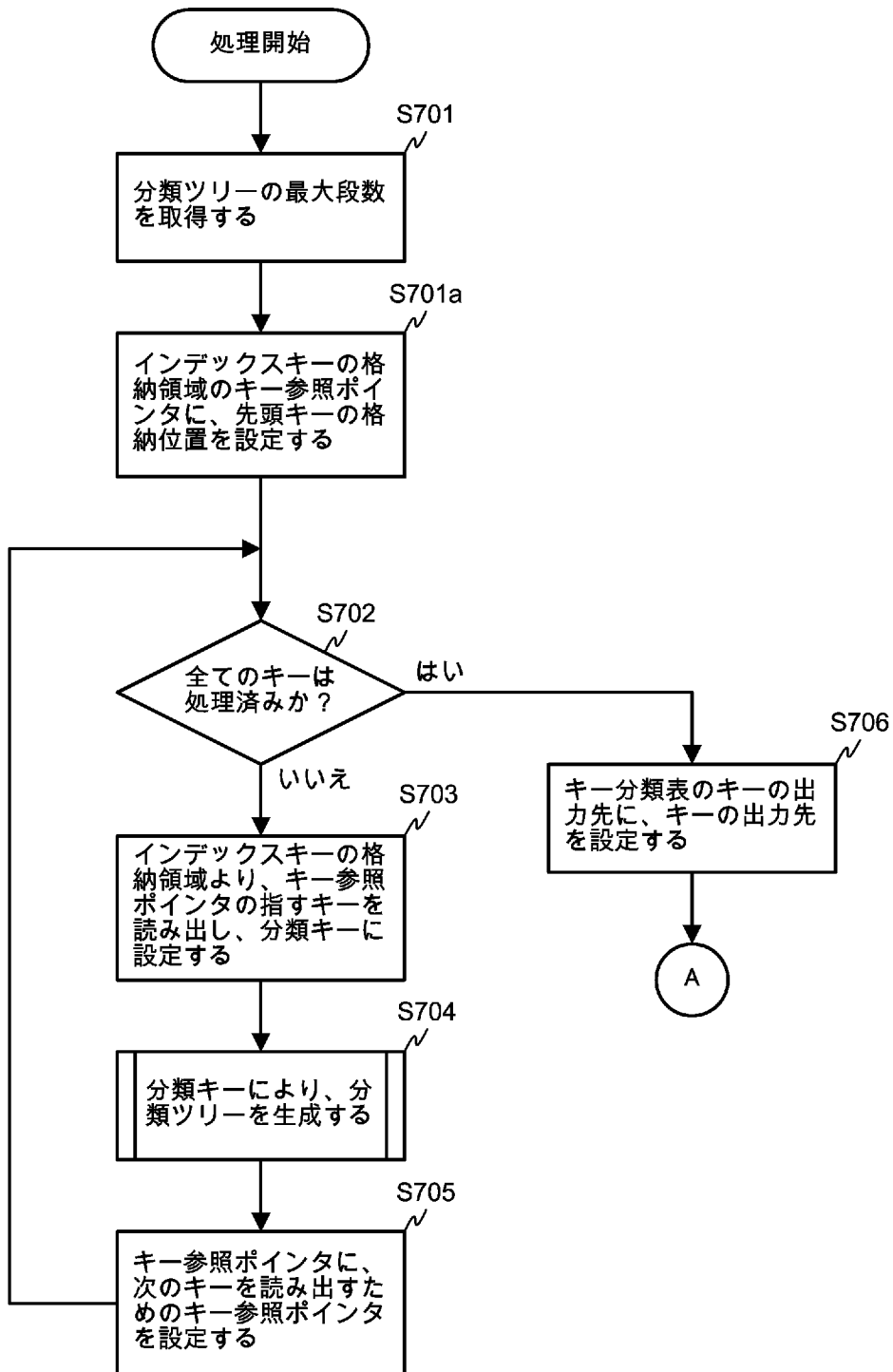
[2]

[2]

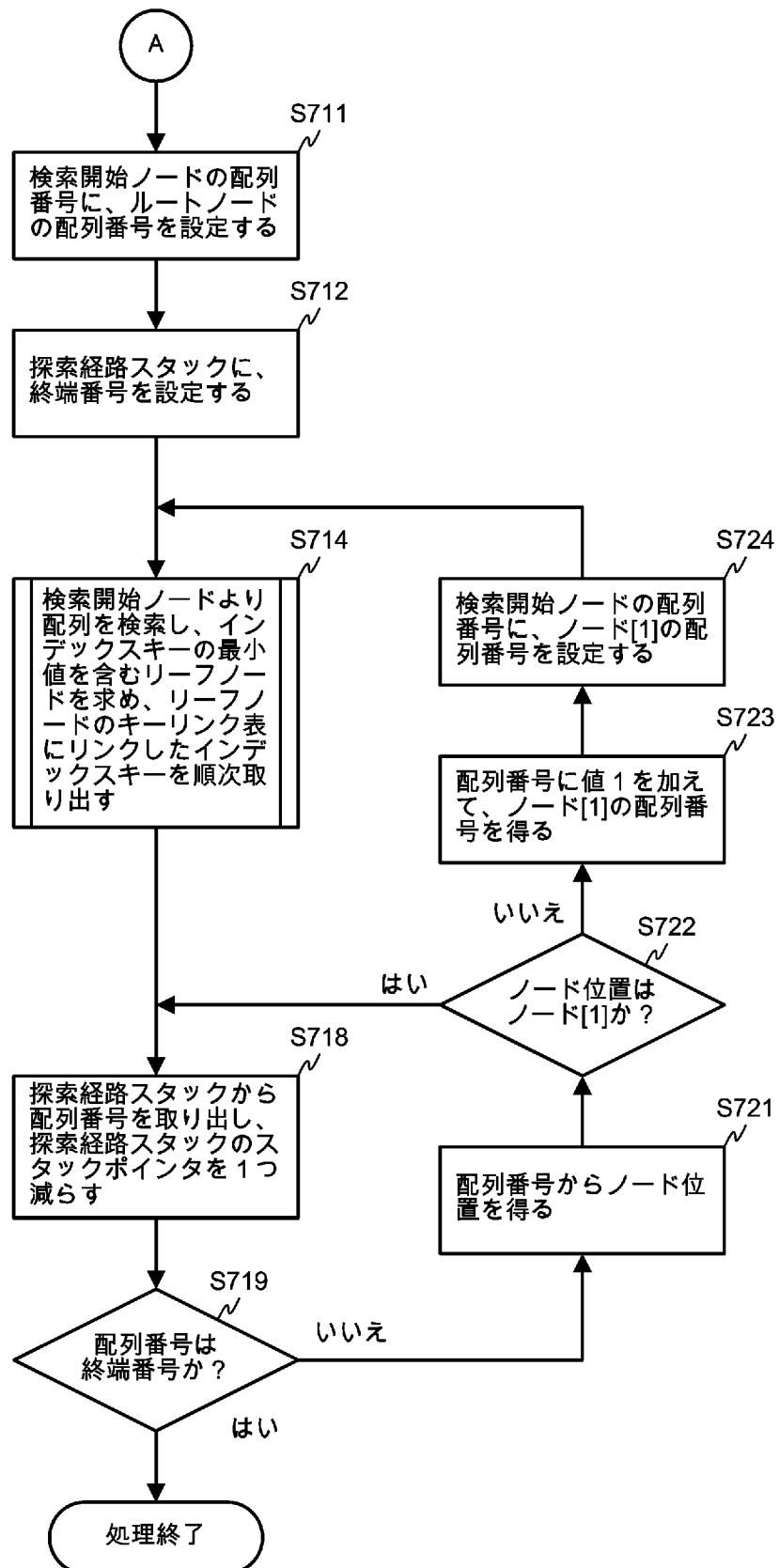
[図6E]



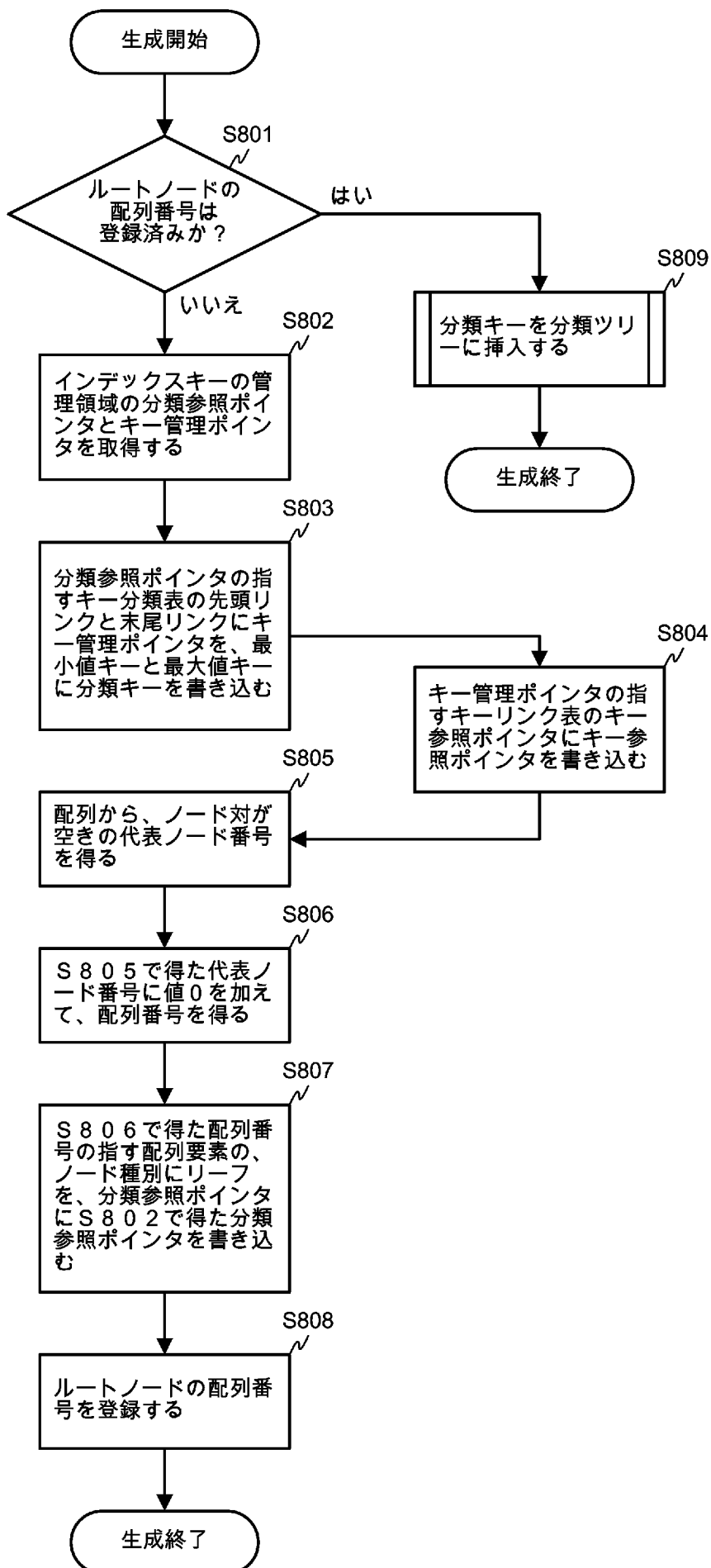
[図7A]



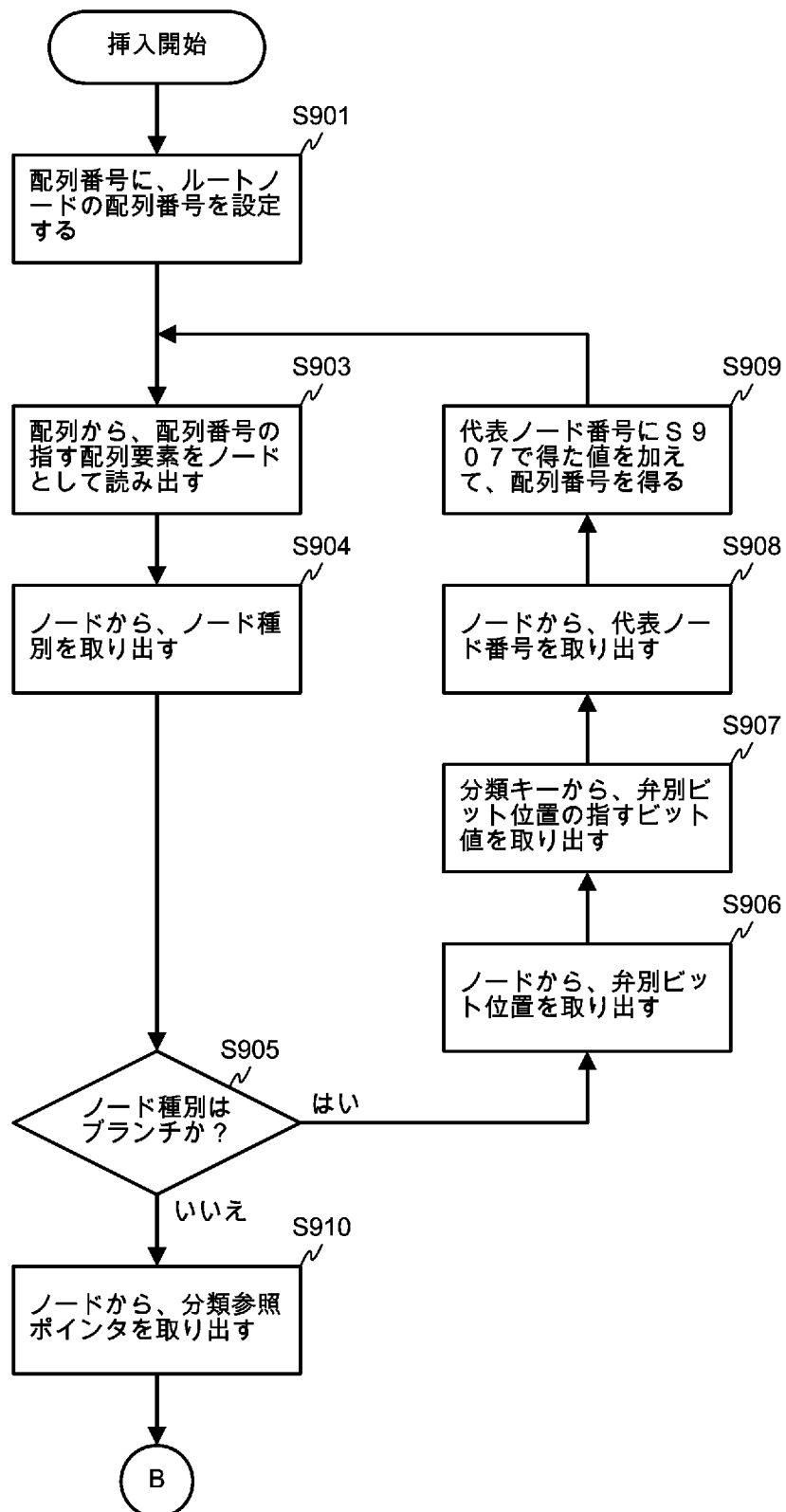
[図7B]



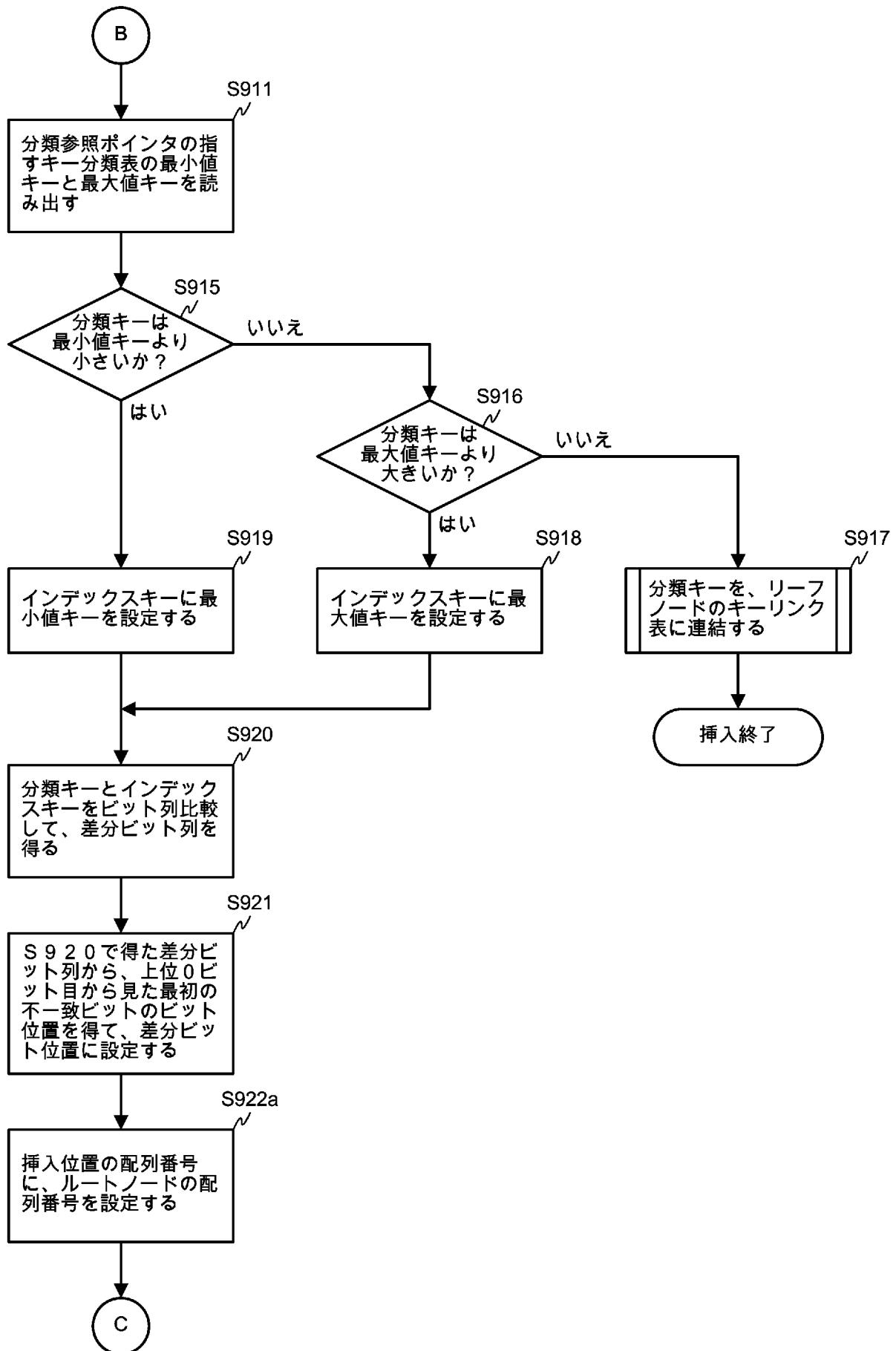
[図8]



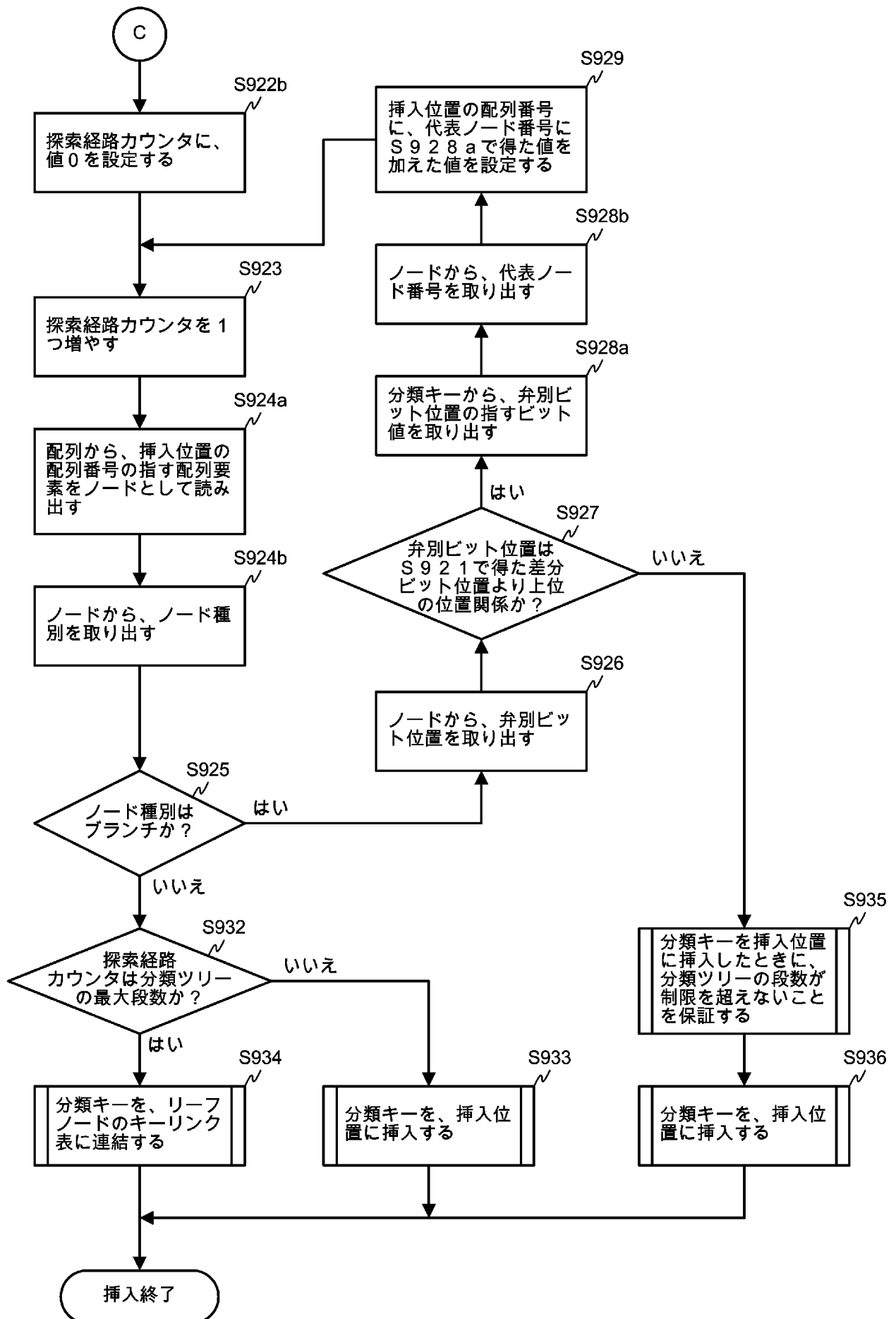
[図9A]



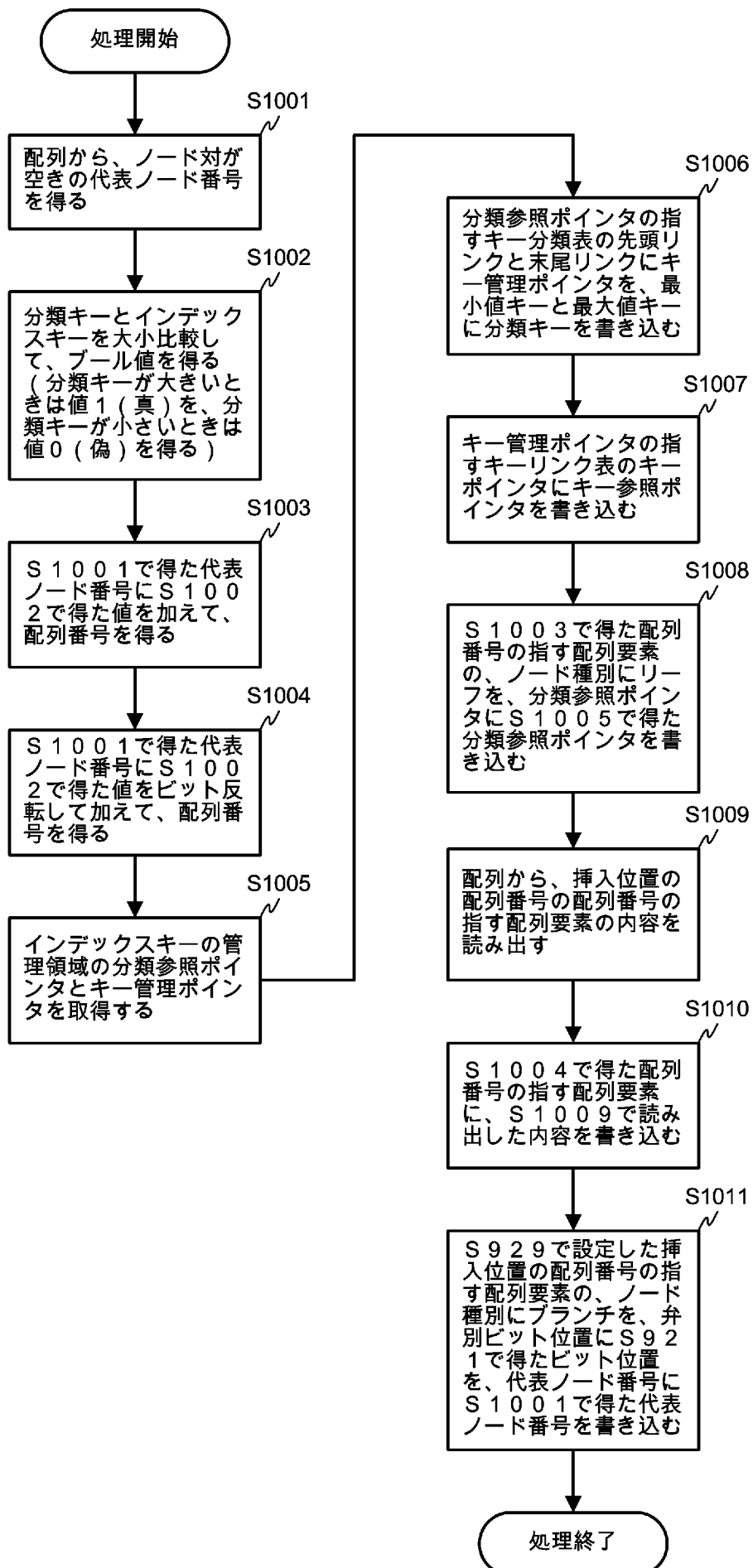
[図9B]



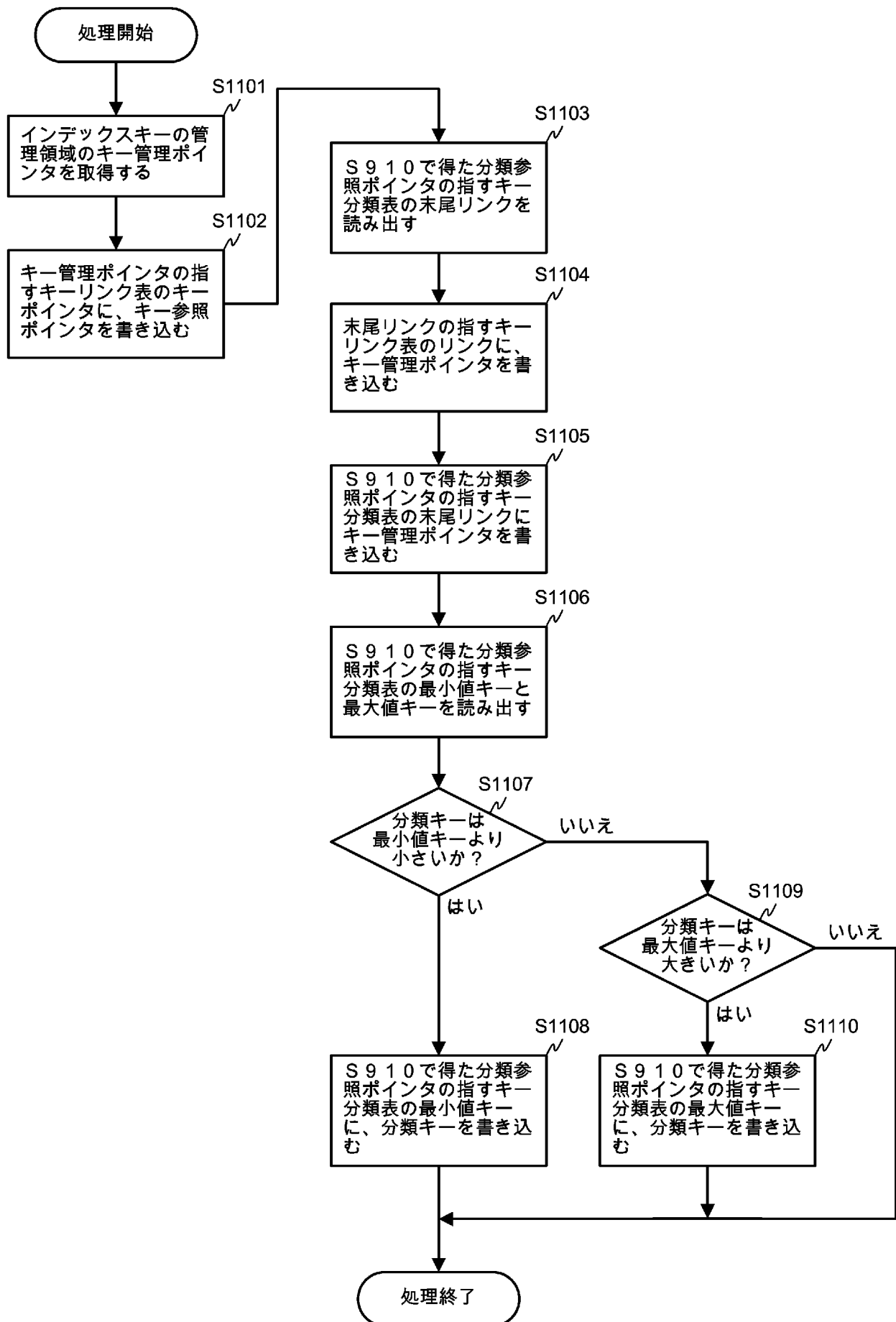
[図9C]



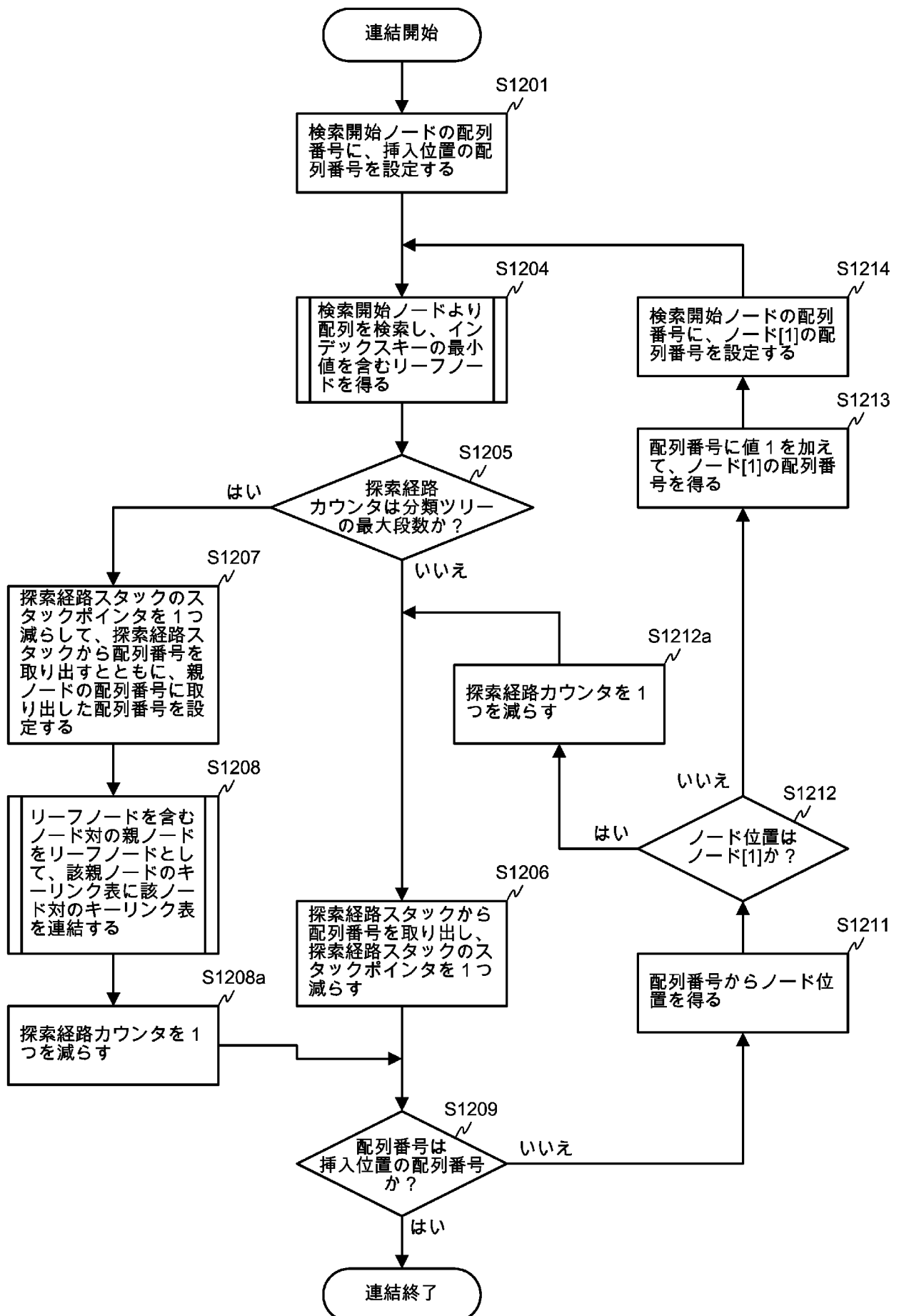
[図10]



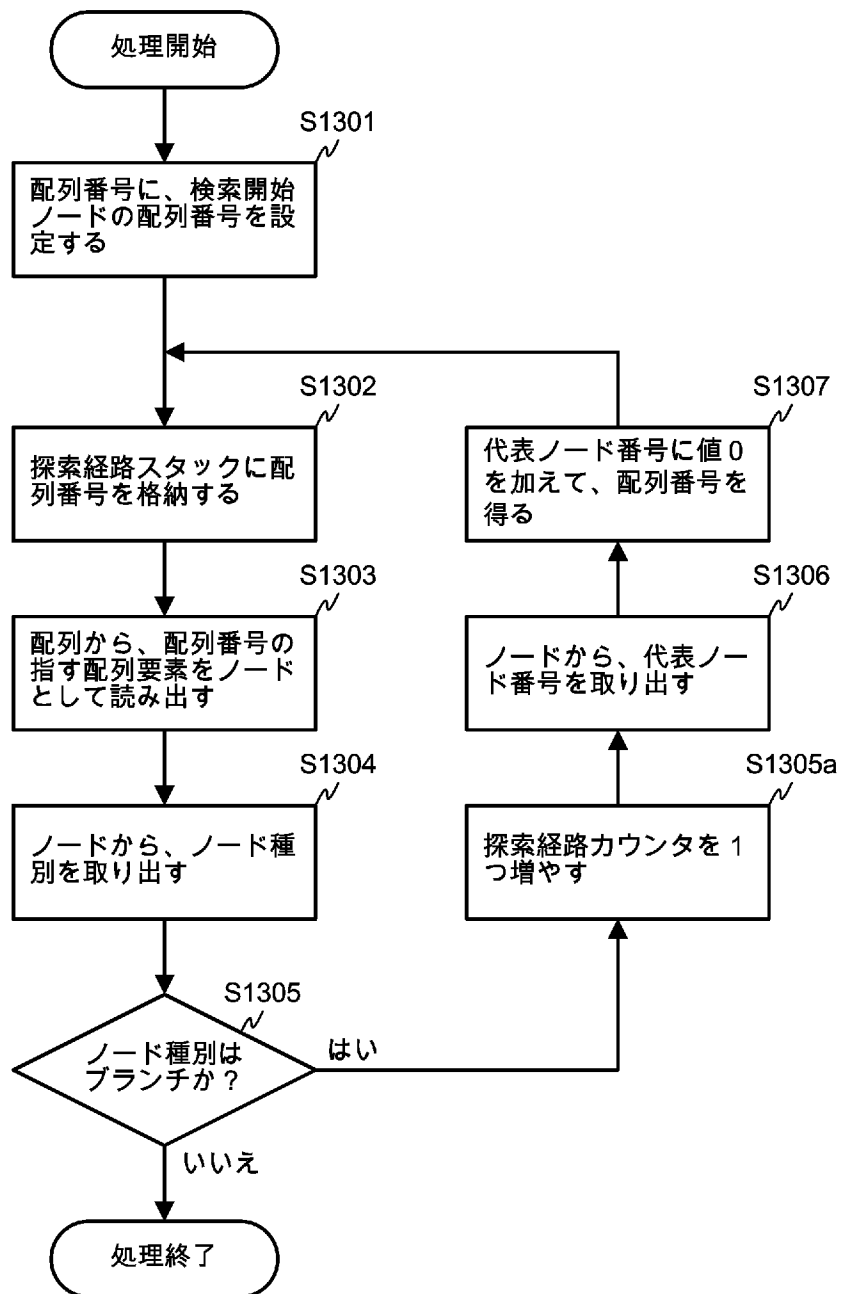
[図11]



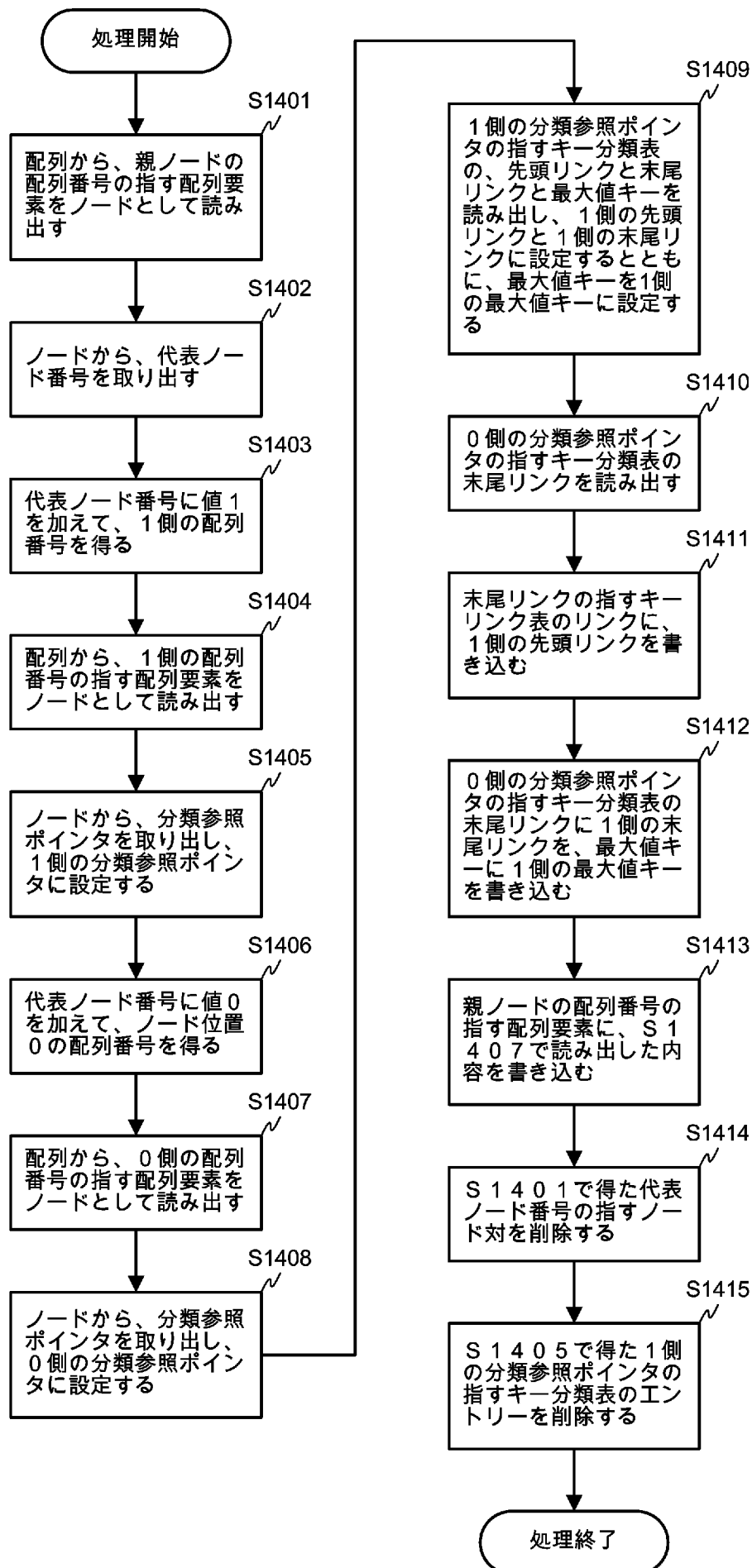
[図12]



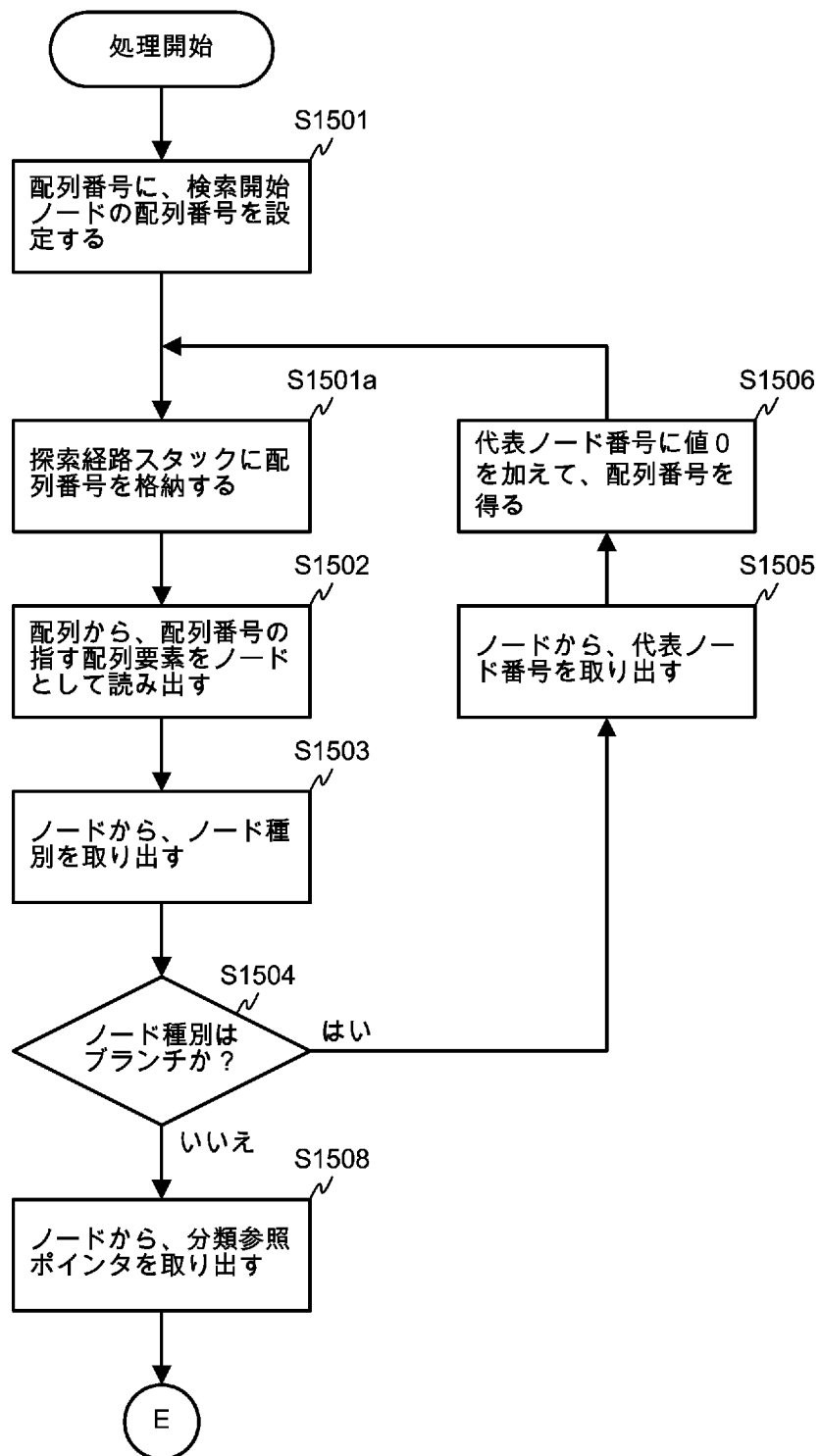
[図13]



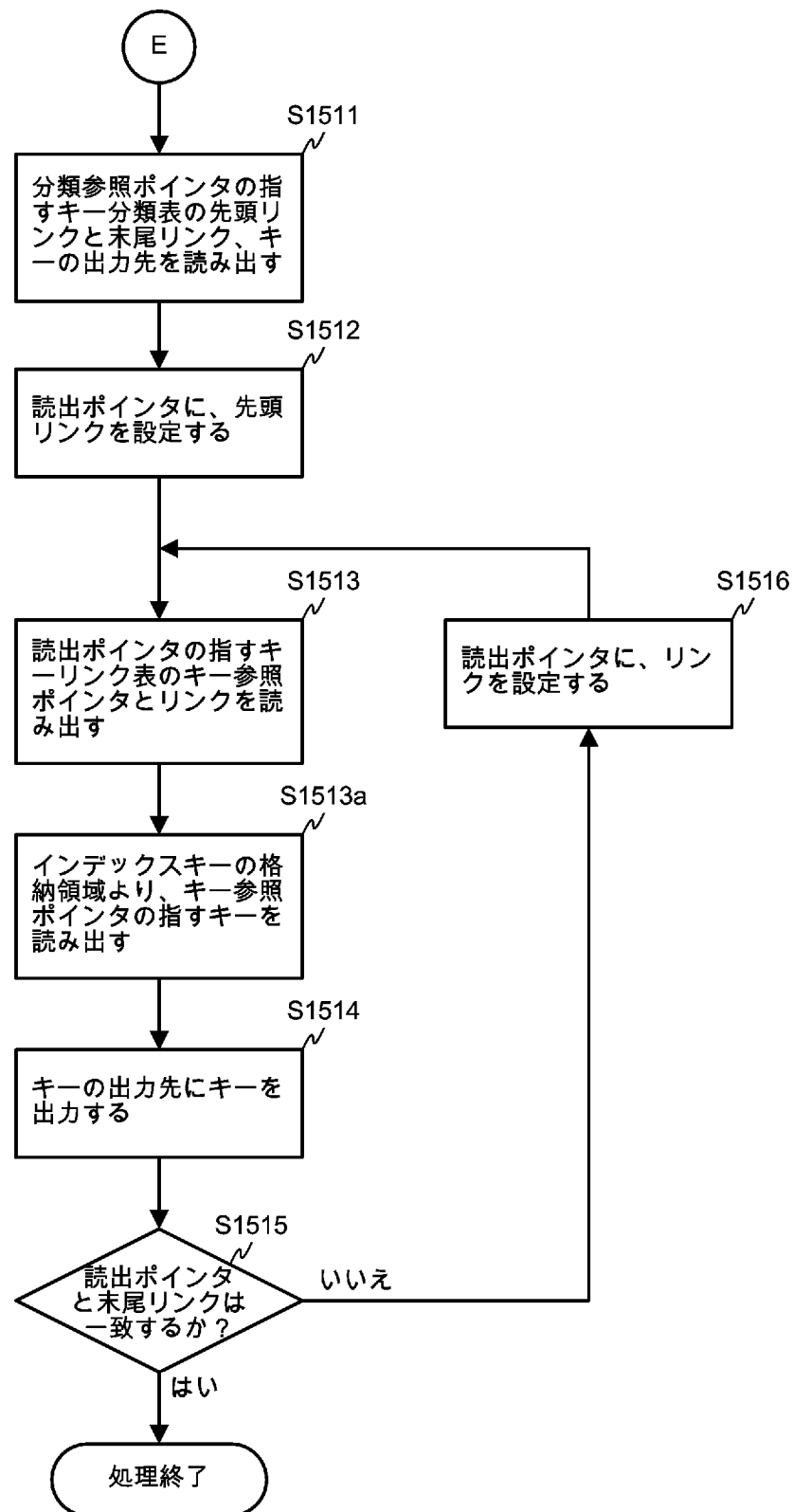
[図14]



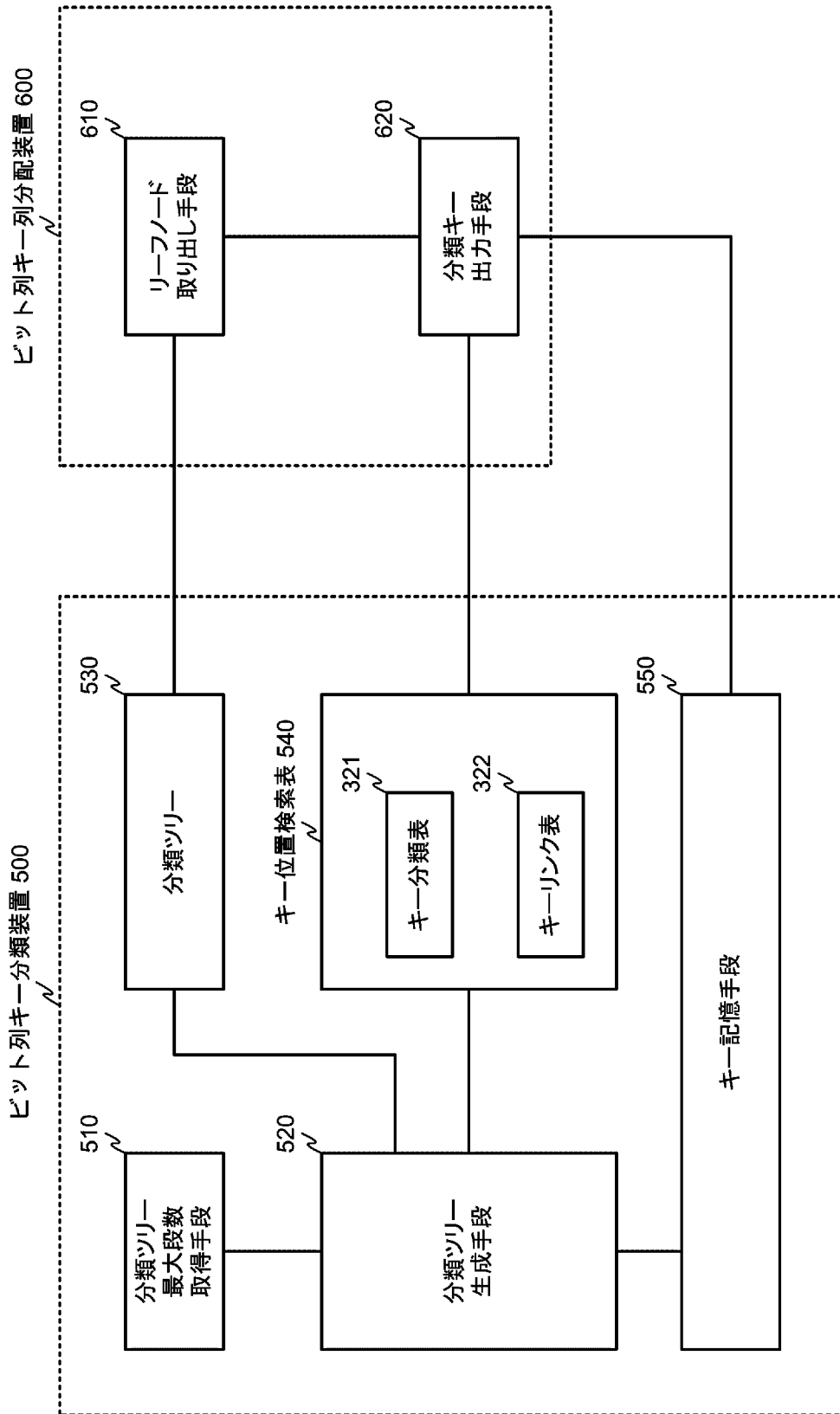
[図15A]



[図15B]



[図16]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2010/006305

A. CLASSIFICATION OF SUBJECT MATTER

G06F17/30 (2006.01) i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F17/30

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Jitsuyo Shinan Koho	1922-1996	Jitsuyo Shinan Toroku Koho	1996-2010
Kokai Jitsuyo Shinan Koho	1971-2010	Toroku Jitsuyo Shinan Koho	1994-2010

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JP 2007-011548 A (Fujitsu Ltd.), 18 January 2007 (18.01.2007), entire text; all drawings & US 2007/0005598 A1	1-22
A	JP 2005-208709 A (Fuji Xerox Co., Ltd.), 04 August 2005 (04.08.2005), entire text; all drawings (Family: none)	1-22
A	JP 2008-269503 A (S. Grants Co., Ltd.), 06 November 2008 (06.11.2008), entire text; all drawings & US 2010/0042596 A1 & US 2010/0042597 A1 & US 2010/0042598 A1 & EP 2149845 A1 & WO 2008/132806 A1	1-22

☒ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

* Special categories of cited documents:

“A” document defining the general state of the art which is not considered to be of particular relevance

“E” earlier application or patent but published on or after the international filing date

“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

“O” document referring to an oral disclosure, use, exhibition or other means

“P” document published prior to the international filing date but later than the priority date claimed

“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

“&” document member of the same patent family

Date of the actual completion of the international search
19 November, 2010 (19.11.10)

Date of mailing of the international search report
30 November, 2010 (30.11.10)

Name and mailing address of the ISA/
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2010/006305

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JP 2008-015872 A (S. Grants Co., Ltd.), 24 January 2008 (24.01.2008), entire text; all drawings & US 2009/0240655 A1 & EP 2048584 A1 & WO 2008/004335 A1 & CN 101484895 A	1-22

A. 発明の属する分野の分類（国際特許分類（I P C））

Int.Cl. G06F17/30(2006.01)i

B. 調査を行った分野

調査を行った最小限資料（国際特許分類（I P C））

Int.Cl. G06F17/30

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	1 9 2 2 - 1 9 9 6 年
日本国公開実用新案公報	1 9 7 1 - 2 0 1 0 年
日本国実用新案登録公報	1 9 9 6 - 2 0 1 0 年
日本国登録実用新案公報	1 9 9 4 - 2 0 1 0 年

国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）

C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求項の番号
A	JP 2007-011548 A（富士通株式会社）2007.01.18, 全文, 全図 & US 2007/0005598 A1	1 - 2 2
A	JP 2005-208709 A（富士ゼロックス株式会社）2005.08.04, 全文, 全図（ファミリーなし）	1 - 2 2
A	JP 2008-269503 A（株式会社エスグランツ）2008.11.06, 全文, 全図 & US 2010/0042596 A1 & US 2010/0042597 A1 & US 2010/0042598 A1 & EP 2149845 A1 & WO 2008/132806 A1	1 - 2 2

☒ C 欄の続きにも文献が列挙されている。☐ パテントファミリーに関する別紙を参照。

* 引用文献のカテゴリー

「A」特に関連のある文献ではなく、一般的技術水準を示すもの
「E」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの
「L」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）
「O」口頭による開示、使用、展示等に言及する文献
「P」国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献

「T」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの

「X」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの

「Y」特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの

「&」同一パテントファミリー文献

国際調査を完了した日

1 9 . 1 1 . 2 0 1 0

国際調査報告の発送日

3 0 . 1 1 . 2 0 1 0

国際調査機関の名称及びあて先

日本国特許庁（I S A / J P）

郵便番号 1 0 0 - 8 9 1 5

東京都千代田区霞が関三丁目4番3号

特許庁審査官（権限のある職員）

吉田 誠

5 M

3 6 5 9

電話番号 0 3 - 3 5 8 1 - 1 1 0 1 内線 3 5 9 9

C (続き) . 関連すると認められる文献		
引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求項の番号
A	JP 2008-015872 A (株式会社エスグランツ) 2008.01.24, 全文, 全図 & US 2009/0240655 A1 & EP 2048584 A1 & WO 2008/004335 A1 & CN 101484895 A	1 - 2 2