



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2024-0103706
(43) 공개일자 2024년07월04일

(51) 국제특허분류(Int. Cl.)
G06F 16/903 (2019.01) G06F 18/22 (2023.01)
(52) CPC특허분류
G06F 16/90344 (2019.01)
G06F 18/22 (2023.01)
(21) 출원번호 10-2022-0186125
(22) 출원일자 2022년12월27일
심사청구일자 2022년12월27일

(71) 출원인
인하대학교 산학협력단
인천광역시 미추홀구 인하로 100(용현동, 인하대학교)
(72) 발명자
심정섭
인천광역시 남동구 남동대로 860, 120동 601호(간석동, 간석래미안자이아파트)
공준호
인천광역시 미추홀구 경인남길102번길 110, 406호(용현동)
김영호
인천광역시 미추홀구 인하로 100, 하이테크센터 1010호(용현동, 인하대학교)
(74) 대리인
양성보

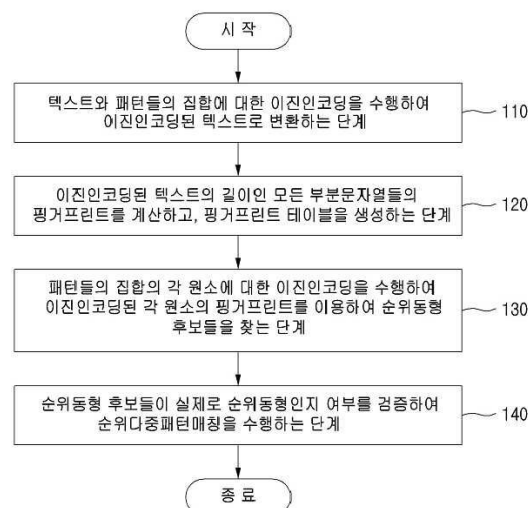
전체 청구항 수 : 총 8 항

(54) 발명의 명칭 이진인코딩을 이용한 순위다중패턴매칭 방법 및 시스템

(57) 요약

이진인코딩을 이용한 순위다중패턴매칭 방법 및 시스템이 제시된다. 본 발명에서 제안하는 이진인코딩을 이용한 순위다중패턴매칭 방법은 텍스트와 패턴들의 집합에 대한 이진인코딩을 수행하여 이진인코딩된 텍스트로 변환하는 단계, 이진인코딩된 텍스트의 길이인 모든 부분문자열들의 핑거프린트를 계산하고, 핑거프린트 테이블을 생성하는 단계, 패턴들의 집합의 각 원소에 대한 이진인코딩을 수행하여 이진인코딩된 각 원소의 핑거프린트를 이용하여 순위동형 후보들을 찾는 단계 및 상기 순위동형 후보들이 실제로 순위동형인지 여부를 검증하여 순위다중패턴매칭을 수행하는 단계를 포함한다.

대표도 - 도1



이 발명을 지원한 국가연구개발사업

과제고유번호	1711152759
과제번호	2020-0-01389-003
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	인공지능융합연구센터지원사업(국고)
연구과제명	[Ezbaro][정부] 인공지능융합연구센터지원(3차년도)
기 여 율	1/1
과제수행기관명	인하대학교 산학협력단
연구기간	2022.01.01 ~ 2022.12.31

명세서

청구범위

청구항 1

텍스트와 패턴들의 집합에 대한 이진인코딩을 수행하여 이진인코딩된 텍스트로 변환하는 단계;

이진인코딩된 텍스트의 길이인 모든 부분문자열들의 핑거프린트를 계산하고, 핑거프린트 테이블을 생성하는 단계;

패턴들의 집합의 각 원소에 대한 이진인코딩을 수행하여 이진인코딩된 각 원소의 핑거프린트를 이용하여 순위동형 후보들을 찾는 단계; 및

상기 순위동형 후보들이 실제로 순위동형인지 여부를 검증하여 순위다중패턴매칭을 수행하는 단계

를 포함하는 순위다중패턴매칭 방법.

청구항 2

제1항에 있어서,

상기 텍스트와 패턴들의 집합에 대한 이진인코딩을 수행하여 이진인코딩된 텍스트로 변환하는 단계는,

길이가 n 인 텍스트 T 와 패턴들의 집합 $\overline{P} = \{P_1, P_2, \dots, P_k\}$ 을 이용하여 $O(n)$ 시간에 이진인코딩을 수행하여 T' 로 변환하는

순위다중패턴매칭 방법.

청구항 3

제2항에 있어서,

상기 이진인코딩된 텍스트의 길이인 모든 부분문자열들의 핑거프린트를 계산하고, 핑거프린트 테이블을 생성하는 단계는,

상기 이진인코딩된 텍스트 T' 의 길이 q 인 모든 부분문자열 q -그램들의 핑거프린트를 계산하고, 핑거프린트 테이블 FT 를 생성하고,

상기 핑거프린트 테이블 FT 는 $O(n + 2^q)$ 공간을 이용하여 $O(n + 2^q)$ 시간에 생성하는

순위다중패턴매칭 방법.

청구항 4

제3항에 있어서,

상기 패턴들의 집합의 각 원소에 대한 이진인코딩을 수행하여 이진인코딩된 각 원소의 핑거프린트를 이용하여 순위동형 후보들을 찾는 단계는,

상기 패턴들의 집합의 각 원소 $P_a (1 \leq a \leq k)$ 에 대하여 위치 테이블을 계산하고, 상기 패턴들의 집합의 각 원소에 대해 이진인코딩한 각 원소 P'_a 의 핑거프린트를 이용하여 순위동형 후보들을 찾기 위해 이진인코딩한 각 원소 P'_a 의 길이 q 인 접두사의 핑거프린트를 계산하는

순위다중패턴매칭 방법.

청구항 5

텍스트와 패턴들의 집합에 대한 이진인코딩을 수행하여 이진인코딩된 텍스트로 변환하고, 이진인코딩된 텍스트의 길이인 모든 부분문자열들의 핑거프린트를 계산하고, 핑거프린트 테이블을 생성하는 텍스트 전처리부; 및 패턴들의 집합의 각 원소에 대한 이진인코딩을 수행하여 이진인코딩된 각 원소의 핑거프린트를 이용하여 순위동형 후보들을 찾고, 상기 순위동형 후보들이 실제로 순위동형인지 여부를 검증하여 순위다중패턴매칭을 수행하는 순위다중패턴 탐색부를 포함하는 순위다중패턴매칭 시스템.

청구항 6

제5항에 있어서,

상기 텍스트 전처리부는,

길이가 n 인 텍스트 T 와 패턴들의 집합 $\overline{P} = \{P_1, P_2, \dots, P_k\}$ 을 이용하여 $O(n)$ 시간에 이진인코딩을 수행하여 T' 로 변환하는

순위다중패턴매칭 시스템.

청구항 7

제6항에 있어서,

상기 텍스트 전처리부는,

상기 이진인코딩된 텍스트 T' 의 길이 q 인 모든 부분문자열 q -그램들의 핑거프린트를 계산하고, 핑거프린트 테이블 FT 를 생성하고,

상기 핑거프린트 테이블 FT 는 $O(n + 2^q)$ 공간을 이용하여 $O(n + 2^q)$ 시간에 생성하는

순위다중패턴매칭 시스템.

청구항 8

제7항에 있어서,

상기 순위다중패턴 탐색부는,

상기 패턴들의 집합의 각 원소 $P_a (1 \leq a \leq k)$ 에 대하여 위치 테이블을 계산하고, 상기 패턴들의 집합의 각 원소에 대해 이진인코딩한 각 원소 P'_a 의 핑거프린트를 이용하여 순위동형 후보들을 찾기 위해 이진인코딩한 각 원소 P'_a 의 길이 q 인 접두사의 핑거프린트를 계산하는

순위다중패턴매칭 시스템.

발명의 설명

기술 분야

[0001] 본 발명은 이진인코딩을 이용한 순위다중패턴매칭 방법 및 시스템에 관한 것이다.

배경 기술

[0002] 대소를 비교할 수 있는 문자로 이루어진 두 문자열 x, y 의 길이가 같고, 같은 위치에 있는 문자의 상대적인 순위가 모두 같으면 x, y 는 순위동형(order-isomorphic)이라 한다. 예를 들어, $x = (5, 9, 4, 7, 2)$ 와 $y = (8, 12, 7, 10, 3)$ 는 순위가 $(3, 5, 2, 4, 1)$ 로 같아 순위동형이다. 순위패턴매칭문제(order-preserving pattern matching problem)는 길이가 n 인 텍스트 T 와 길이가 m 인 패턴 P 가 주어졌을 때, P 와 순위동형인

T 의 모든 부분문자열들을 찾는 문제이다. 순위패턴매칭문제는 KMP 알고리즘을 이용하여 $O(n+m \log m)$ 시간에 해결하는 알고리즘이 제시되었다. q -그램과 Horspool 알고리즘을 기반으로 $O(mn+nq \log q+q!)$ 시간에 순위패턴매칭문제를 해결하는 알고리즘이 제시되었고 이진인코딩을 이용하여 $O(nm+2^q)$ 시간에 순위패턴매칭문제를 해결하는 알고리즘이 제시되었다.

[0003] 순위다중패턴매칭문제(order-preserving multiple pattern matching problem)는 길이가 n 인 텍스트 T 와 패턴들의 집합 $\overline{P} = \{P_1, P_2, \dots, P_k\}$ 가 주어졌을 때, P_a ($1 \leq a \leq k$)와 순위동형인 T 의 모든 부분문자열들을 찾는 문제이다. 순위다중패턴매칭문제는 $\overline{P}$ 에서 가장 짧은 패턴의 길이를 m , 가장 긴 패턴의 길이를 $\overline{m}$, 패턴들의 길이의 합을 M 이라 할 때, Aho-Corasick 알고리즘을 기반으로 $O(n \log M)$ 시간에 해결하는 알고리즘이 제시되었다. M 이 m 에 대한 다항식일 때, $O(n)$ 시간에 탐색을 수행하는 알고리즘과 $O((n/m) \log M)$ 시간에 탐색을 수행하는 알고리즘이 제시되었다. $O(q!+nq \log q)$ 시간에 T 의 q -그램들에 대한 핑거프린트 정보를 생성하고, 이를 이용하여 $O(M \log \overline{m} + kq \log q + knm)$ 시간에 탐색을 수행하는 알고리즘이 제시되었다. 이후, 순위다중패턴매칭문제를 Z-함수를 이용하여 $O(n+\overline{m})$ 시간에 해결하는 병렬 알고리즘, 해싱을 기반으로 해결하는 공간효율적인 알고리즘, Karp-Rabin 알고리즘을 이용한 병렬 알고리즘 등이 제시되었다.

[0004] 그 외에도 패턴매칭문제를 효율적으로 해결하기 위한 방법으로 압축패턴매칭 알고리즘, 알고리즘 병렬화 등 많은 연구가 진행되고 있다.

[0005] 순위패턴매칭문제와 순위다중패턴매칭문제는 주가지수, 음악 멜로디 등 시계열 데이터 분석에 활용될 수 있다.

선행기술문헌

비특허문헌

- [0006] (비특허문헌 0001) [1] M. Kubica, T. Kulczynski, J. Radoszewski, W. Rytter, T. Walen, "A linear time algorithm for consecutive permutation pattern matching," Information Processing Letters, Vol. 113, pp. 430-433, 2013.
- (비특허문헌 0002) [2] J. Kim, P. Eades, R. Fleischer, S.-H. Hong, C. S. Iliopoulos, K. Park, S. J. Puglisi, T. Tokuyama, "Order-preserving matching," Theoretical Computer Science, Vol. 525, pp. 68-79, 2014.
- (비특허문헌 0003) [3] S. Cho, J. Na, K. Pa, J. Sim, "A fast algorithm for order-preserving pattern matching," Information Processing Letters, Vol. 115, pp. 397-402, 2015.
- (비특허문헌 0004) [4] T. Chhabra, J. Tariho, "A filtration method for order-preserving matching," Information Processing Letters, Vol. 116, pp. 71-74, 2016.
- (비특허문헌 0005) [5] 김영준, 김영호, 심정섭, "이진인코딩을 이용한 Horspool 알고리즘 기반 순위패턴매칭 알고리즘," 한국정보과학회 학술발표논문집, pp. 2112-2214, 2018.
- (비특허문헌 0006) [6] 유광모, 김영호, 심정섭, "텍스트의 핑거프린트를 이용한 순위다중패턴매칭 알고리즘," 한국정보과학회 학술발표논문집, pp. 1214-1216, 2018.
- (비특허문헌 0007) [7] H. Cormen, E. Leiserson, L. Rivest, C. Stein, Introduction to Algorithms, 3rd Ed., The MIT press, 2009.

발명의 내용

해결하려는 과제

[0007] 본 발명이 이루고자 하는 기술적 과제는 이진인코딩된 텍스트의 팽거프린트를 이용하여 $O(M(\log \bar{m} + kn) + 2^q)$ 시간에 순위다중패턴매칭문제를 해결하는 이진인코딩을 이용한 순위다중패턴매칭 방법 및 시스템을 제공하는데 있다. 또한, 이진인코딩을 이용한 순위다중패턴매칭 방법 및 시스템은 $O(n + 2^q)$ 공간을 이용하여 $O(n + 2^q)$ 시간에 텍스트를 전처리하고, $O(\bar{m}(\log \bar{m} + n))$ 시간에 탐색 과정을 수행하는 병렬 구현 방법을 제안한다. $O(k)$ 개의 스레드를 사용하여 $O(\bar{m}(\log \bar{m} + n))$ 시간에 패턴 탐색 과정을 병렬적으로 구현한다.

과제의 해결 수단

[0008] 일 측면에 있어서, 본 발명에서 제안하는 이진인코딩을 이용한 순위다중패턴매칭 방법은 텍스트와 패턴들의 집합에 대한 이진인코딩을 수행하여 이진인코딩된 텍스트로 변환하는 단계, 이진인코딩된 텍스트의 길이인 모든 부분문자열들의 팽거프린트를 계산하고, 팽거프린트 테이블을 생성하는 단계, 패턴들의 집합의 각 원소에 대한 이진인코딩을 수행하여 이진인코딩된 각 원소의 팽거프린트를 이용하여 순위동형 후보들을 찾는 단계 및 상기 순위동형 후보들이 실제로 순위동형인지 여부를 검증하여 순위다중패턴매칭을 수행하는 단계를 포함한다.

[0009] 상기 텍스트와 패턴들의 집합에 대한 이진인코딩을 수행하여 이진인코딩된 텍스트로 변환하는 단계는 길이가 n 인 텍스트 T 와 패턴들의 집합 $\bar{P} = \{P_1, P_2, \dots, P_k\}$ 을 이용하여 $O(n)$ 시간에 이진인코딩을 수행하여 T' 로 변환한다.

[0010] 상기 이진인코딩된 텍스트의 길이인 모든 부분문자열들의 팽거프린트를 계산하고, 팽거프린트 테이블을 생성하는 단계는 상기 이진인코딩된 텍스트 T' 의 길이 q 인 모든 부분문자열 q -그램들의 팽거프린트를 계산하고, 팽거프린트 테이블 FT 를 생성하고, 상기 팽거프린트 테이블 FT 는 $O(n + 2^q)$ 공간을 이용하여 $O(n + 2^q)$ 시간에 생성한다.

[0011] 상기 패턴들의 집합의 각 원소에 대한 이진인코딩을 수행하여 이진인코딩된 각 원소의 팽거프린트를 이용하여 순위동형 후보들을 찾는 단계는 상기 패턴들의 집합의 각 원소 $P_a (1 \leq a \leq k)$ 에 대하여 위치 테이블을 계산하고, 상기 패턴들의 집합의 각 원소에 대해 이진인코딩한 각 원소 P'_a 의 팽거프린트를 이용하여 순위동형 후보들을 찾기 위해 이진인코딩한 각 원소 P'_a 의 길이 q 인 접두사의 팽거프린트를 계산한다.

[0012] 또 다른 일 측면에 있어서, 본 발명에서 제안하는 이진인코딩을 이용한 순위다중패턴매칭 시스템은 텍스트와 패턴들의 집합에 대한 이진인코딩을 수행하여 이진인코딩된 텍스트로 변환하고, 이진인코딩된 텍스트의 길이인 모든 부분문자열들의 팽거프린트를 계산하고, 팽거프린트 테이블을 생성하는 텍스트 전처리부, 패턴들의 집합의 각 원소에 대한 이진인코딩을 수행하여 이진인코딩된 각 원소의 팽거프린트를 이용하여 순위동형 후보들을 찾고, 상기 순위동형 후보들이 실제로 순위동형인지 여부를 검증하여 순위다중패턴매칭을 수행하는 순위다중패턴 탐색부를 포함한다.

발명의 효과

[0013] 본 발명의 실시예들에 따르면 이진인코딩된 텍스트의 팽거프린트를 이용하여 $O(M(\log \bar{m} + kn) + 2^q)$ 시간에 순위다중패턴매칭문제를 해결할 수 있다. 또한, 이진인코딩을 이용한 순위다중패턴매칭 방법 및 시스템은 $O(n + 2^q)$ 공간을 이용하여 $O(n + 2^q)$ 시간에 텍스트를 전처리하고, $O(\bar{m}(\log \bar{m} + n))$ 시간에 탐색 과정을 수행하는 병렬 구현 방법을 제안하며, $O(k)$ 개의 스레드를 사용하여 $O(\bar{m}(\log \bar{m} + n))$ 시간에 패턴 탐색 과정을 병렬적으로 구현할 수 있다. 실험 결과, q 가 커질수록 제안한 알고리즘이 기존 알고리즘보다 수행시간이 느려지지만 같은 q 에 대하여 공간 효율적으로 수행되고, 제시한 병렬 구현 방법은 기존 알고리즘보다 공간효율적 이면서 수행시간은 유사하다.

도면의 간단한 설명

- [0014] 도 1은 본 발명의 일 실시예에 따른 이진인코딩을 이용한 순위다중패턴매칭 방법을 설명하기 위한 흐름도이다.
- 도 2는 본 발명의 일 실시예에 따른 텍스트의 이진인코딩에 대한 핑거프린트 테이블 생성 과정을 설명하기 위한 도면이다.
- 도 3은 본 발명의 일 실시예에 따른 핑거프린트 테이블 생성 알고리즘을 나타내는 도면이다.
- 도 4는 본 발명의 일 실시예에 따른 이진인코딩을 이용한 순위다중패턴매칭 알고리즘을 나타내는 도면이다.
- 도 5는 본 발명의 일 실시예에 따른 이진인코딩을 이용한 순위다중패턴매칭 시스템의 구성을 나타내는 도면이다.
- 도 6은 본 발명의 일 실시예에 따른 각 알고리즘의 랜덤에 대해 발생하는 핑거프린트의 개수를 나타내는 그래프이다.
- 도 7은 본 발명의 일 실시예에 따른 각 알고리즘의 전압에 대해 발생하는 핑거프린트의 개수를 나타내는 그래프이다.
- 도 8은 본 발명의 일 실시예에 따른 각 알고리즘의 snp에 대해 발생하는 핑거프린트의 개수를 나타내는 그래프이다.

발명을 실시하기 위한 구체적인 내용

- [0015] 순위다중패턴매칭문제는 길이가 n 인 텍스트 T 와 패턴들의 집합 $\bar{P} = \{P_1, P_2, \dots, P_k\}$ 가 주어졌을 때, $\bar{P}$ 에 속한 패턴들과 순위동형인 T 의 모든 부분문자열들을 찾는 문제이다. q 개의 연속적인 문자인 q -그램, $\bar{P}$ 에서 가장 짧은 패턴의 길이를 m , 가장 긴 패턴의 길이를 $\bar{m}$, 모든 패턴들의 길이의 합을 M 이라 할 때, q -그램과 계승수체계를 이용하여 $O(M \log \bar{m} + (k+n)q \log q + knm + q!)$ 시간에 순위다중패턴매칭문제를 해결하는 알고리즘이 제시되었다.
- [0016] 본 발명에서는 이진인코딩된 텍스트의 핑거프린트를 이용하여 $O(M(\log \bar{m} + kn) + 2^q)$ 시간에 순위다중패턴매칭문제를 해결하는 알고리즘을 제안한다. 또한, 제시한 알고리즘은 $O(n + 2^q)$ 공간을 이용하여 $O(n + 2^q)$ 시간에 텍스트를 전처리하고, $O(\bar{m}(\log \bar{m} + n))$ 시간에 탐색 과정을 수행하는 병렬 구현 방법을 제시한다. $O(k)$ 개의 스레드를 사용하여 $O(M(\log \bar{m} + kn))$ 시간에 패턴 탐색 과정을 병렬적으로 구현한다. 실험 결과, q 가 커질수록 제안한 알고리즘이 기존 알고리즘보다 수행시간이 느려지지만 같은 q 에 대하여 공간 효율적으로 수행된다. 제시한 병렬 구현 방법은 기존 알고리즘보다 공간효율적이면서 수행시간은 유사하다. 이하, 본 발명의 실시 예를 첨부된 도면을 참조하여 상세하게 설명한다.
- [0018] 먼저, 본 발명의 실시예에 따른 용어에 관하여 설명한다.
- [0019] 알파벳 Σ 는 대소 비교가 가능한 문자로 이루어진 집합이다. Σ 의 문자들로 구성된 문자열을 Σ^* 로 표기하고 $x(\in \Sigma^*)$ 의 길이를 $|x|$ 로 표현한다. x 의 $i(0 \leq i < |x|)$ 번째 문자는 $x[i]$ 로 표기하고, x 의 i 번째부터 j 번째까지의 부분문자열(substring)은 $x[i..j](0 \leq i \leq j < |x|)$ 로 표기한다. 두 문자열 $x, y(\in \Sigma^*)$ 가 순위동형이면 $x \approx y$ 로 표기한다. 패턴들의 집합 $\bar{P} = \{P_1, P_2, \dots, P_k\}(P_i \in \Sigma^*, 1 \leq i \leq k)$ 에서 가장 짧은 패턴의 길이를 m , 가장 긴 패턴의 길이를 $\bar{m}$, 모든 패턴의 길이의 합을 M 으로 표기한다.
- [0020] 문자열 $x(\in \Sigma^*)$ 는 최근접이웃표현법(nearest neighbor representation)과 접두사표현법(prefix

representation)을 이용하여 나타낼 수 있다. x 의 최근접이웃표현법은 다음과 같이 정의된다[1,2].

$$LMin_x[i] = j \quad \text{if } x[j] = \min\{x[k] : k \in [0, i-1], x[k] \geq x[i]\}$$

$$LMax_x[i] = j \quad \text{if } x[j] = \max\{x[k] : k \in [0, i-1], x[k] \leq x[i]\}$$

만약 $LMin_x[i]$ (또는 $LMax_x[i]$)에 위의 조건을 만족하는 인덱스 j 가 없다면 -1 을 저장하고, 조건을 만족하는 j 가 둘 이상이면 가장 큰 인덱스를 저장한다. x 의 접두사표현법은 다음과 같이 정의된다[2].

$$\mu_x[i] = |\{j : x[j] \leq x[i], 0 \leq j < i\}|$$

<표 1>

i	0	1	2	3	4	5	6
$x[i]$	6	7	4	8	17	11	9
$LMin_x[i]$	-1	-1	0	-1	-1	4	5
$LMax_x[i]$	-1	0	-1	1	3	3	3
$\mu_x[i]$	0	1	0	3	4	4	4

<표 1>은 문자열 $x = (6, 7, 4, 8, 17, 11, 9)$ 에 대한 순위표현법 중 위치테이블($LMin_x$, $LMax_x$)과 접두사테이블(μ_x)을 보여준다. 이 테이블들은 모두 순위통계트리(order statistic tree)를 이용하여 $O(|x|\log|x|)$ 시간에 계산할 수 있다. 길이가 같은 두 문자열 $x, y (\in \Sigma^*)$ 의 순위동형 여부는 x 또는 y 에 대한 위치테이블을 이용하여 $O(|x|)$ 시간에 판단할 수 있다[1,2].

도 1은 본 발명의 일 실시예에 따른 이진인코딩을 이용한 순위다중패턴매칭 방법을 설명하기 위한 흐름도이다.

제안하는 이진인코딩을 이용한 순위다중패턴매칭 방법은 텍스트와 패턴들의 집합에 대한 이진인코딩을 수행하여 이진인코딩된 텍스트로 변환하는 단계(110), 이진인코딩된 텍스트의 길이인 모든 부분문자열들의 핑거프린트를 계산하고, 핑거프린트 테이블을 생성하는 단계(120), 패턴들의 집합의 각 원소에 대한 이진인코딩을 수행하여 이진인코딩된 각 원소의 핑거프린트를 이용하여 순위동형 후보들을 찾는 단계(130) 및 상기 순위동형 후보들이 실제로 순위동형인지 여부를 검증하여 순위다중패턴매칭을 수행하는 단계(140)를 포함한다.

단계(110)에서, 텍스트와 패턴들의 집합에 대한 이진인코딩을 수행하여 이진인코딩된 텍스트로 변환한다. 길이가 n 인 텍스트 T 와 패턴들의 집합 $\bar{P} = \{P_1, P_2, \dots, P_k\}$ 을 이용하여 $O(n)$ 시간에 이진인코딩을 수행하여 T' 로 변환한다.

단계(120)에서, 이진인코딩된 텍스트의 길이인 모든 부분문자열들의 핑거프린트를 계산하고, 핑거프린트 테이블을 생성한다. 상기 이진인코딩된 텍스트 T' 의 길이 q 인 모든 부분문자열 q -그램들의 핑거프린트를 계산하고, 핑거프린트 테이블 FT 를 생성한다. 상기 핑거프린트 테이블 FT 는 $O(n + 2^q)$ 공간을 이용하여 $O(n + 2^q)$ 시간에 생성할 수 있다.

단계(130)에서, 패턴들의 집합의 각 원소에 대한 이진인코딩을 수행하여 이진인코딩된 각 원소의 핑거프린트를 이용하여 순위동형 후보들을 찾는다. 상기 패턴들의 집합의 각 원소 $P_a (1 \leq a \leq k)$ 에 대하여 위치 테이블을 계산하고, 상기 패턴들의 집합의 각 원소에 대해 이진인코딩한 각 원소 P'_a 의 핑거프린트를 이용하여 순위동형 후보들을 찾기 위해 이진인코딩한 각 원소 P'_a 의 길이 q 인 접두사의 핑거프린트, 즉 $f'(P'_a[0..q-1])$ 을 계산한다. 이때 $q < m$ 으로 가정한다.

[0034] 단계(140)에서, 상기 순위동형 후보들이 실제로 순위동형인지 여부를 검증하여 순위다중패턴매칭을 수행한다.

[0035] 상기 $f'(P_a'[0..q-1]) = C$ 라면 $FT[C]$ 를 확인한다. $FT[C] = \{X_1, X_2, \dots, X_t\}$ 라면, P_a 와 $T[X_i \dots X_i + |P_a| - 1] (1 \leq i \leq t)$ 이 순위동형인지 여부를 P_a 의 위치테이블을 이용하여 검증한다. 탐색 과정은 위치테이블들을 계산하기 위해 총 $O(M \log \bar{m})$ 시간, 핑거프린트 계산에 총 $O(kq)$ 시간, 순위동형 검증에 최악의 경우 총 $O(knM)$ 시간이 소요된다. 따라서 탐색 과정은 총 $O(M(\log \bar{m} + kn))$ 시간이 소요되고, 전체 알고리즘은 $O(M(\log \bar{m} + kn) + 2^q)$ 시간에 수행된다.

[0037] 도 2는 본 발명의 일 실시예에 따른 텍스트의 이진인코딩에 대한 핑거프린트 테이블 생성 과정을 설명하기 위한 도면이다.

[0038] 연속된 q 개의 문자인 q -그램에 대한 핑거프린트[3]를 이용하여 순위다중패턴매칭문제를 해결하는 알고리즘이 제시되었다[6]. q -그램 x 의 핑거프린트 $f(x)$ 를 다음과 같이 정의된다[3,6].

$$f(x) = \sum_{k=0}^{q-1} \mu_x[k] \times k!.$$

[0039]

[0040] 핑거프린트를 이용한 순위다중패턴매칭 알고리즘[6]은 다음과 같이 진행된다. 텍스트의 모든 q -그램들에 대한 핑거프린트 정보를 저장한 테이블 FT 를 생성한다. $FT[A]$ 는 q -그램 $x = T[i..i+q-1]$ 의 핑거프린트 $f(x)$ 가 $A (0 \leq A \leq q!)$ 일 때, x 의 시작위치 i 를 모두 저장한다. 이후 패턴의 q -그램의 핑거프린트와 텍스트 q -그램의 핑거프린트가 일치하면 패턴의 위치테이블을 이용하여 순위동형을 검증한다. FT 는 $O(n + q!)$ 공간을 사용하여 $O(nq \log q + q!)$ 시간에 생성할 수 있다.

[0041] 이진인코딩[4]을 이용하여 q -그램의 핑거프린트를 계산하고 순위패턴매칭문제를 해결하는 알고리즘이 제시되었다[5]. 이 알고리즘에서 문자열 T 의 이진인코딩 T' 은 다음과 같이 정의된다.

$$T'[i] (0 \leq i < |T| - 1) = \begin{cases} 1 & \text{if } T[i] < T[i+1], \\ 0 & \text{otherwise} \end{cases}$$

[0042]

[0043] 예를 들어, $T = (6, 7, 4, 8, 17, 11, 9, 12, 15, 2, 5)$ 이면, $T' = (1, 0, 1, 1, 0, 0, 1, 1, 0, 1)$ 이다(도 2 참조).

[0044] 이진인코딩된 문자열의 q -그램 x 에 대한 핑거프린트 $f'(x)$ 는 다음과 같이 정의된다.

$$f'(x) = \sum_{k=0}^{q-1} x[k] \times 2^{(q-1)-k} \quad [7]. \quad (1)$$

[0045]

[0046] 도 2의 예에서 $q = 3$ 이면 $f'(T'[0..2]) = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 5$ 이고,

[0047] $f'(T'[1..3]) = 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 3$ 이다.

[0049] 도 3은 본 발명의 일 실시예에 따른 핑거프린트 테이블 생성 알고리즘을 나타내는 도면이다.

[0050] 본 발명에서는 텍스트 T 와 패턴 집합 $\bar{P} = \{P_1, P_2, \dots, P_k\}$ 가 주어졌을 때, T 의 이진인코딩 T' 의 핑거프린트를 이용하여 순위다중패턴매칭문제를 해결하는 알고리즘을 제시한다. 먼저 텍스트 T 를 T' 으로 변환하고, T' 의

모든 q -그램에 대한 핑거프린트 정보를 저장한 핑거프린트 테이블 FT 를 생성한다. 마지막으로, $P_a(1 \leq a \leq k)$ 를 이진인코딩한 P'_a 의 핑거프린트를 이용하여 순위동형 후보들을 찾은 뒤, 이 후보들이 실제로 순위동형인지 여부를 검증하여 순위다중패턴매칭문제를 해결한다.

[0051] 구체적 절차는 다음과 같다. 먼저, T 를 $O(n)$ 시간에 이진인코딩[4]을 이용하여 T' 으로 변환한다(도 3의 1번 줄부터 6번 줄). 이후, T' 의 길이 q 인 모든 부분문자열, 즉 q -그램들의 핑거프린트를 식(1)을 이용하여 계산한다. $f'(T'[i..i+q-1])$ 을 계산했을 때, $f'(T'[i+1..i+q])$ 는 Horner의 방법[7]을 이용하여 $O(1)$ 시간에 계산할 수 있다(도 3의 12번째 줄). $f'(T'[i..i+q-1]) = B$ 이면, $FT[B]$ 에 i 를 추가한다. 예를 들어, 도 2의 예제와 같이 $T = (6, 7, 4, 8, 17, 11, 9, 12, 15, 2, 5)$ 일 때, T' 의 각 q -그램들은 $FT[5], FT[3], FT[6], FT[4], FT[1], FT[3], FT[6], FT[5]$ 에 삽입된다. FT 는 $O(n+2^q)$ 공간을 이용하여 $O(n+2^q)$ 시간에 생성할 수 있다.

[0053] 도 4는 본 발명의 일 실시예에 따른 이진인코딩을 이용한 순위다중패턴매칭 알고리즘을 나타내는 도면이다.

[0054] T 에서 $P_a(1 \leq a \leq k)$ 의 탐색은 다음과 같이 진행된다. 먼저, P_a 의 위치테이블들을 계산한다. 그리고 순위동형 후보들을 찾기 위해 P'_a 의 길이 q 인 접두사의 핑거프린트, 즉 $f'(P'_a[0..q-1])$ 을 계산한다. 이때 $q < m$ 으로 가정한다. $f'(P'_a[0..q-1]) = C$ 라면 $FT[C]$ 를 확인한다. $FT[C] = \{X_1, X_2, \dots, X_t\}$ 라면, P_a 와 $T[X_i \dots X_i + |P_a| - 1](1 \leq i \leq t)$ 이 순위동형인지 여부를 P_a 의 위치테이블을 이용하여 검증한다(도 4의 6번 줄부터 9번 줄). 탐색 과정은 위치테이블들을 계산하기 위해 총 $O(M \log \bar{m})$ 시간, 핑거프린트 계산에 총 $O(kq)$ 시간, 순위동형 검증에 최악의 경우 총 $O(knM)$ 시간이 소요된다. 따라서 탐색 과정은 총 $O(M(\log \bar{m} + kn))$ 시간이 소요되고, 전체 알고리즘은 $O(M(\log \bar{m} + kn) + 2^q)$ 시간에 수행된다.

[0056] 도 5는 본 발명의 일 실시예에 따른 이진인코딩을 이용한 순위다중패턴매칭 시스템의 구성을 나타내는 도면이다.

[0058] 본 실시예에 따른 이진인코딩을 이용한 순위다중패턴매칭 시스템(500)은 프로세서(510), 버스(520), 네트워크 인터페이스(530), 메모리(540) 및 데이터베이스(550)를 포함할 수 있다. 메모리(540)는 운영체제(541) 및 이진인코딩을 이용한 순위다중패턴매칭 루틴(542)을 포함할 수 있다. 프로세서(510)는 텍스트 전처리부(511) 및 순위다중패턴 탐색부(512)를 포함할 수 있다. 다른 실시예들에서 이진인코딩을 이용한 순위다중패턴매칭 시스템(500)은 도 5의 구성요소들보다 더 많은 구성요소들을 포함할 수도 있다. 그러나, 대부분의 종래기술적 구성요소들을 명확하게 도시할 필요성은 없다. 예를 들어, 이진인코딩을 이용한 순위다중패턴매칭 시스템(500)은 디스플레이나 트랜시버(transceiver)와 같은 다른 구성요소들을 포함할 수도 있다.

[0059] 메모리(540)는 컴퓨터에서 판독 가능한 기록 매체로서, RAM(random access memory), ROM(read only memory) 및 디스크 드라이브와 같은 비소멸성 대용량 기록장치(permanent mass storage device)를 포함할 수 있다. 또한, 메모리(540)에는 운영체제(541)와 이진인코딩을 이용한 순위다중패턴매칭 루틴(542)을 위한 프로그램 코드가 저장될 수 있다. 이러한 소프트웨어 구성요소들은 드라이브 메커니즘(drive mechanism, 미도시)을 이용하여 메모리(540)와는 별도의 컴퓨터에서 판독 가능한 기록 매체로부터 로딩될 수 있다. 이러한 별도의 컴퓨터에서 판독 가능한 기록 매체는 플로피 드라이브, 디스크, 테이프, DVD/CD-ROM 드라이브, 메모리 카드 등의 컴퓨터에서 판독 가능한 기록 매체(미도시)를 포함할 수 있다. 다른 실시예에서 소프트웨어 구성요소들은 컴퓨터에서 판독 가능한 기록 매체가 아닌 네트워크 인터페이스(530)를 통해 메모리(540)에 로딩될 수도 있다.

- [0060] 버스(520)는 이진인코딩을 이용한 순위다중패턴매칭 시스템(500)의 구성요소들간의 통신 및 데이터 전송을 가능하게 할 수 있다. 버스(520)는 고속 시리얼 버스(high-speed serial bus), 병렬 버스(parallel bus), SAN(Storage Area Network) 및/또는 다른 적절한 통신 기술을 이용하여 구성될 수 있다.
- [0061] 네트워크 인터페이스(530)는 이진인코딩을 이용한 순위다중패턴매칭 시스템(500)을 컴퓨터 네트워크에 연결하기 위한 컴퓨터 하드웨어 구성요소일 수 있다. 네트워크 인터페이스(530)는 이진인코딩을 이용한 순위다중패턴매칭 시스템(500)을 무선 또는 유선 커넥션을 통해 컴퓨터 네트워크에 연결시킬 수 있다.
- [0062] 데이터베이스(550)는 이진인코딩을 이용한 순위다중패턴매칭을 위해 필요한 모든 정보를 저장 및 유지하는 역할을 할 수 있다. 도 5에서는 이진인코딩을 이용한 순위다중패턴매칭 시스템(500)의 내부에 데이터베이스(550)를 구축하여 포함하는 것으로 도시하고 있으나, 이에 한정되는 것은 아니며 시스템 구현 방식이나 환경 등에 따라 생략될 수 있고 혹은 전체 또는 일부의 데이터베이스가 별개의 다른 시스템 상에 구축된 외부 데이터베이스로서 존재하는 것 또한 가능하다.
- [0063] 프로세서(510)는 기본적인 산술, 로직 및 이진인코딩을 이용한 순위다중패턴매칭 시스템(500)의 입출력 연산을 수행함으로써, 컴퓨터 프로그램의 명령을 처리하도록 구성될 수 있다. 명령은 메모리(540) 또는 네트워크 인터페이스(530)에 의해, 그리고 버스(520)를 통해 프로세서(510)로 제공될 수 있다. 프로세서(510)는 텍스트 전처리부(511) 및 순위다중패턴 탐색부(512)를 위한 프로그램 코드를 실행하도록 구성될 수 있다. 이러한 프로그램 코드는 메모리(540)와 같은 기록 장치에 저장될 수 있다.
- [0064] 텍스트 전처리부(511) 및 순위다중패턴 탐색부(512)는 도 1의 단계들(110~140)을 수행하기 위해 구성될 수 있다.
- [0065] 이진인코딩을 이용한 순위다중패턴매칭 시스템(500)은 텍스트 전처리부(511) 및 순위다중패턴 탐색부(512)를 포함할 수 있다.
- [0066] 텍스트 전처리부(511)는 텍스트와 패턴들의 집합에 대한 이진인코딩을 수행하여 이진인코딩된 텍스트로 변환한다. 길이가 n 인 텍스트 T 와 패턴들의 집합 $\overline{P} = \{P_1, P_2, \dots, P_k\}$ 을 이용하여 $O(n)$ 시간에 이진인코딩을 수행하여 T' 로 변환한다.
- [0067] 텍스트 전처리부(511)는 이진인코딩된 텍스트의 길이인 모든 부분문자열들의 핑거프린트를 계산하고, 핑거프린트 테이블을 생성한다. 상기 이진인코딩된 텍스트 T' 의 길이 q 인 모든 부분문자열 q -그램들의 핑거프린트를 계산하고, 핑거프린트 테이블 FT 를 생성한다. 상기 핑거프린트 테이블 FT 는 $O(n+2^q)$ 공간을 이용하여 $O(n+2^q)$ 시간에 생성할 수 있다.
- [0068] 순위다중패턴 탐색부(512)는 패턴들의 집합의 각 원소에 대한 이진인코딩을 수행하여 이진인코딩된 각 원소의 핑거프린트를 이용하여 순위동형 후보들을 찾는다. 상기 패턴들의 집합의 각 원소 $P_a (1 \leq a \leq k)$ 에 대하여 위치 테이블을 계산하고, 상기 패턴들의 집합의 각 원소에 대해 이진인코딩한 각 원소 P'_a 의 핑거프린트를 이용하여 순위동형 후보들을 찾기 위해 이진인코딩한 각 원소 P'_a 의 길이 q 인 접두사의 핑거프린트, 즉 $f'(P'_a[0..q-1])$ 을 계산한다. 이때 $q < m$ 으로 가정한다.
- [0069] 순위다중패턴 탐색부(512)는 상기 순위동형 후보들이 실제로 순위동형인지 여부를 검증하여 순위다중패턴매칭을 수행한다.
- [0070] 상기 $f'(P'_a[0..q-1]) = C$ 라면 $FT[C]$ 를 확인한다. $FT[C] = \{X_1, X_2, \dots, X_t\}$ 라면, P_a 와 $T[X_i \dots X_i + |P_a| - 1] (1 \leq i \leq t)$ 이 순위동형인지 여부를 P_a 의 위치테이블을 이용하여 검증한다. 탐색 과정은 위치테이블들을 계산하기 위해 총 $O(M \log \overline{m})$ 시간, 핑거프린트 계산에 총 $O(kq)$ 시간, 순위동형 검증에 최악의 경우 총 $O(knM)$ 시간이 소요된다. 따라서 탐색 과정은 총 $O(M(\log \overline{m} + kn))$ 시간이 소요되고, 전체 알고리즘은 $O(M(\log \overline{m} + kn) + 2^q)$ 시간에 수행된다.

[0071] 본 발명의 실시예에 따른 실험 환경은 다음과 같다. OS는 Windows 10(64bit), CPU는 AMD Ryzen 9 5950X (3.4GHz), RAM은 64GB, 개발 툴은 Visual Studio 2022, 프로그래밍 언어는 C++를 사용하였다. 본 논문에서 제시한 알고리즘과 기존의 알고리즘[6]을 $q = \{4, 5, 6, 7\}$ 에 대해 실험하였고, 기존 알고리즘[6]과 마찬가지로 평거프린트를 2개 이용하여 순위동형 후보를 찾았다.

[0072] 한편, 본 발명에서 제시한 알고리즘의 탐색 과정은 $O(k)$ 개의 스레드를 이용하여 병렬적으로 계산할 수 있다. 각 스레드 $i (1 \leq i \leq k)$ 는 병렬적으로 P_i 에 대한 탐색 과정을 수행하여 모든 패턴의 위치테이블들 계산에 $O(\overline{m} \log \overline{m})$ 시간, 평거프린트 계산에 $O(q)$ 시간, 순위동형 검증에 최악의 경우 $O(n\overline{m})$ 시간이 소요된다.

[0073] 본 발명에서 제시한 알고리즘의 순차 구현을 BF , 위에서 설명한 병렬 구현을 PBF 라 하고, 기존 알고리즘 [6]의 구현을 TP 라 하자. 입력 데이터의 종류와 패턴의 길이를 변화시키며 100번 실험한 평균 시간을 비교하였다. 단, 문자열을 이진인코딩하면 길이가 1 줄어들게 되므로, BF , PBF 의 q -그램의 길이는 TP 의 q -그램의 길이보다 1만큼 짧다.

[0074] 텍스트의 길이 $n = 1,000,000$, 패턴의 개수 $k = 1,000$ 일 때, $m = \{10, 15, 20\}$ 에 대해 랜덤 데이터, 전압 데이터, S&P 500 지수 데이터에 대해 알고리즘 수행시간을 비교하였다(표2, 표3, 표4 참조).

[0075] <표 2>

q	$q!$	2^{q-1}	m	Input data		
				random		
				BF	PBF	TP
4	24	8	10	842	92	387
			15	855	88	386
			20	884	89	386
5	120	16	10	475	61	81
			15	474	56	83
			20	483	57	84
6	720	32	10	271	42	20
			15	259	39	20
			20	251	39	20
7	5,040	64	10	192	35	15
			15	137	31	15
			20	136	29	15

[0076]

[0077] <표 3>

q	$q!$	2^{q-1}	m	Input data		
				voltage		
				BF	PBF	TP
4	24	8	10	745	66	454
			15	761	67	447
			20	771	66	460
5	120	16	10	384	42	141
			15	384	41	141
			20	384	41	143
6	720	32	10	203	30	43
			15	190	28	43
			20	189	28	45
7	5,040	64	10	124	26	21
			15	101	24	23
			20	101	24	23

[0078]

<표 4>

q	$q!$	2^{q-1}	m	Input data		
				snp		
				BF	PBF	TP
4	24	8	10	748	68	462
			15	766	66	463
			20	761	67	469
5	120	16	10	379	43	154
			15	381	41	155
			20	377	41	156
6	720	32	10	203	30	54
			15	190	28	53
			20	189	28	54
7	5,040	64	10	132	24	27
			15	108	23	28
			20	105	23	28

랜덤 데이터 *random* 은 $|\Sigma| = 1,000,000$ 인 집합에 대해 무작위로 생성된 데이터이다. 전압 데이터 *voltage* 는 프랑스 쏘(Sceaux)에 위치한 어느 가정의 분당 평균전압을 2006년 12월 16일부터 2008년 12월 2일까지 측정한 데이터이다. S&P 500지수 데이터 *snp* 는 2010년 1월 4일부터 2021년 8월 27일까지의 S&P 500지수의 일별 종가 데이터이다. 실험 결과, $q = \{4, 5\}$ 일 때, 모든 입력 데이터에서 *PBF*, *TP*, *BF* 순으로 빠르게 수행되었다. *random*에서는 $q = \{6, 7\}$, *voltage*에서는 $q = 7$ 일 때, *TP*가 가장 빠르게 수행되었다. 텍스트에서 발생하는 모든 핑거프린트 중 패턴의 핑거프린트와 일치하는 텍스트 부분문자열에 대해 패턴과 순위동형을 검증한다. 즉, 패턴과 동일한 핑거프린트의 수와 알고리즘의 수행시간은 밀접한 관련이 있다.

도 6은 본 발명의 일 실시예에 따른 각 알고리즘의 랜덤에 대해 발생하는 핑거프린트의 개수를 나타내는 그래프이다.

도 7은 본 발명의 일 실시예에 따른 각 알고리즘의 전압에 대해 발생하는 핑거프린트의 개수를 나타내는 그래프이다.

도 8은 본 발명의 일 실시예에 따른 각 알고리즘의 snp에 대해 발생하는 핑거프린트의 개수를 나타내는 그래프이다.

도 6, 도 7 및 도 8은 $q = 4$ 일 때, *BF*와 *TP*에 대해 각 데이터에서 발생하는 핑거프린트의 개수를 나타낸다. x 축은 핑거프린트, y 축은 핑거프린트의 개수이다. *BF*는 이진인코딩된 문자열에 대해 핑거프린트를 계산하기 때문에, 동일한 크기의 q 에 대해 실험하였을 때 발생할 수 있는 핑거프린트의 총 수는 *BF*, *PBF*는 2^{q-1} 개이고 *TP*는 $q!$ 개이다. 동일한 q 에 대해서 *TP*의 핑거프린트 수($q!$)가 *BF*의 핑거프린트 수(2^{q-1})보다 많아 핑거프린트가 일치할 확률이 줄어들게 되므로 *TP*가 *BF*보다 빠르게 수행되었다. q 가 커질수록 *TP*와 *BF*의 수행시간 차이가 커지지만, 사용하는 공간의 크기 차이도 커진다. 즉, *BF*는 *TP*와 비교하여 같은 q 에 대하여 공간 효율적으로 수행된다. *PBF*는 실제 시계열 데이터에서도 *TP*와 수행시간이 유사하면서도 공간사용량이 적다.

이상에서 설명된 장치는 하드웨어 구성요소, 소프트웨어 구성요소, 및/또는 하드웨어 구성요소 및 소프트웨어 구성요소의 조합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성요소는, 예를 들어, 프로세서, 콘트롤러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴

퓨터, FPGA(field programmable gate array), PLU(programmable logic unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(OS) 및 상기 운영 체제 상에서 수행되는 하나 이상의 소프트웨어 애플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설명된 경우도 있지만, 해당 기술분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소(processing element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 컨트롤러를 포함할 수 있다. 또한, 병렬 프로세서(parallel processor)와 같은, 다른 처리 구성(processing configuration)도 가능하다.

[0089] 소프트웨어는 컴퓨터 프로그램(computer program), 코드(code), 명령(instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로(collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성요소(component), 물리적 장치, 가상 장치(virtual equipment), 컴퓨터 저장 매체 또는 장치에 구체화(embodiment)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.

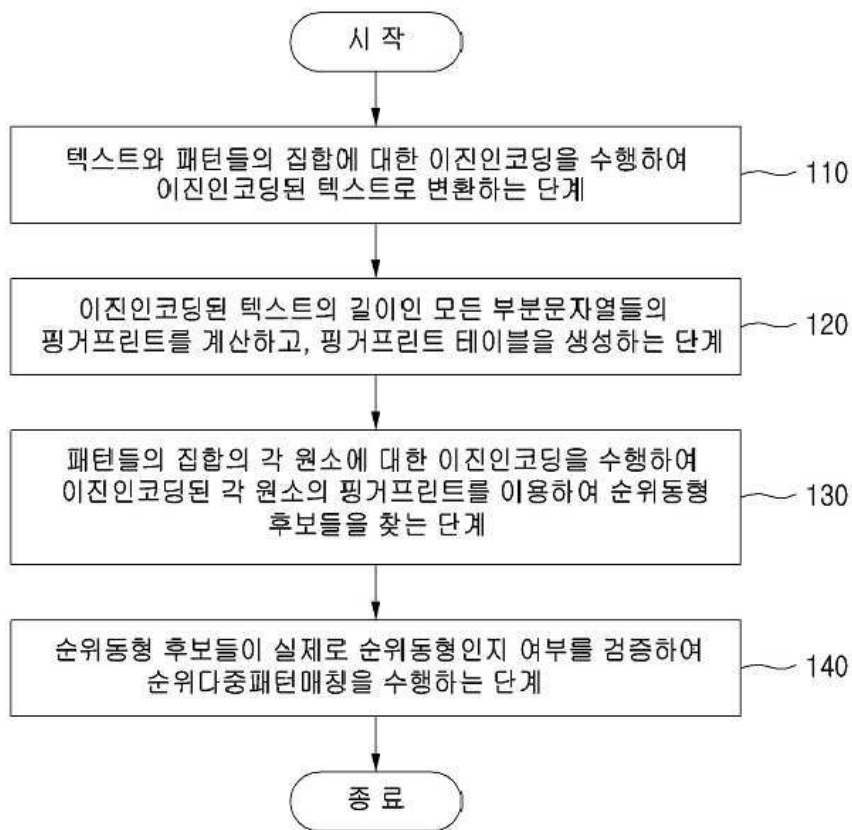
[0090] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 실시예를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다.

[0091] 이상과 같이 실시예들이 비록 한정된 실시예와 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 상기의 기재로부터 다양한 수정 및 변형이 가능하다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다.

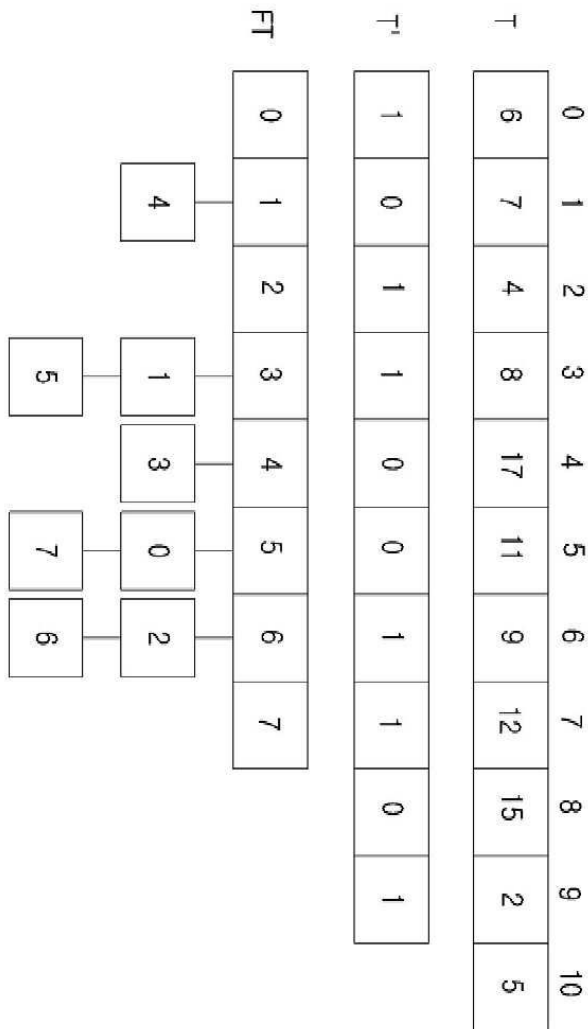
[0092] 그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.

도면

도면1



도면2



도면3

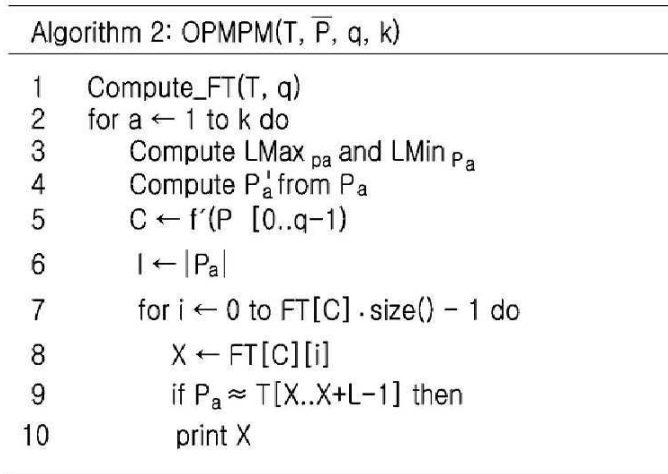
Algorithm 1: Compute_FT(T, q)

```

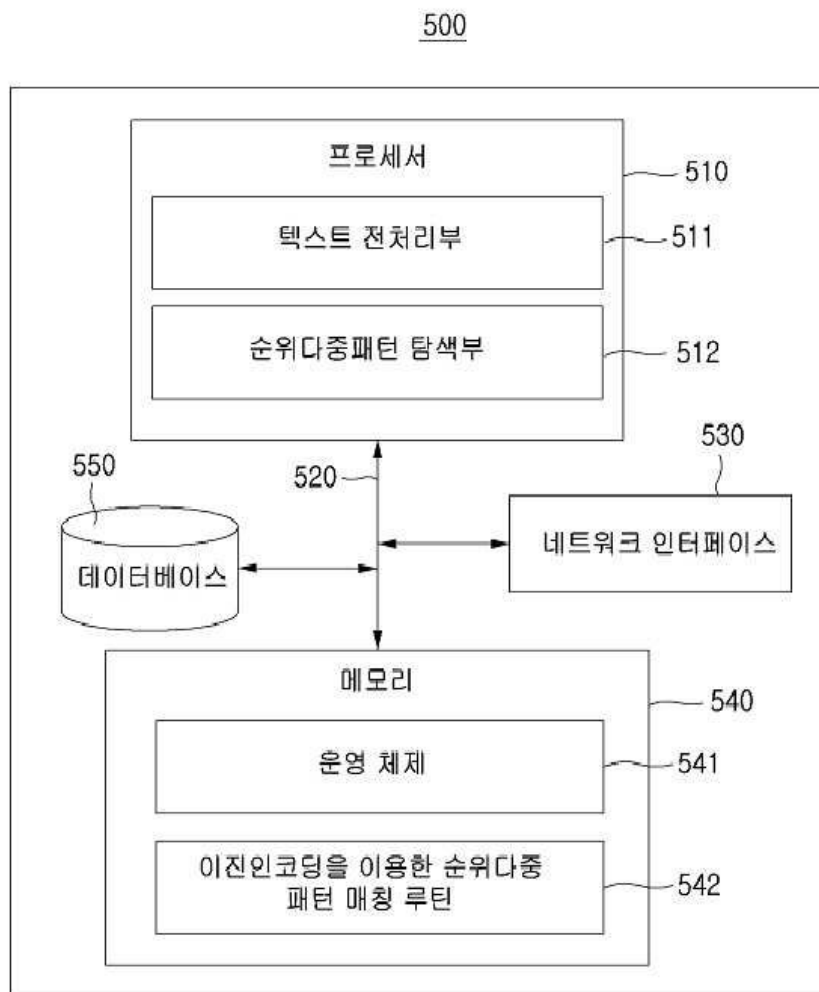
1  n ← |T|
2  for i ← 0 to n-2 do
3    if T[i] < T[i+1] then
4      T'[i] ← 1
5    else
6      T'[i] ← 0
7  B ← 0
8  for i ← 0 to q-1 do
9    B ← B + T'[i] × 2(q-1)-i
10 FT[B].push(i)
11 for i ← 1 to (n-1)-q do
12   B ← B × 2 + T'[i+q-1] - T'[i-1] × 2q
13   FT[B].push(i)

```

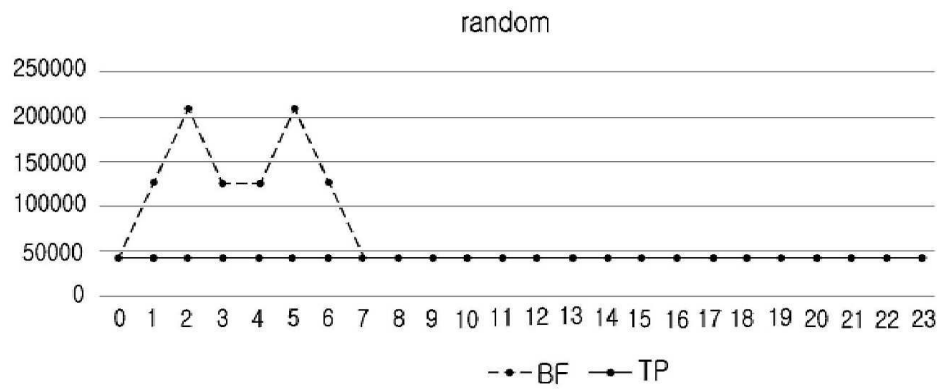
도면4



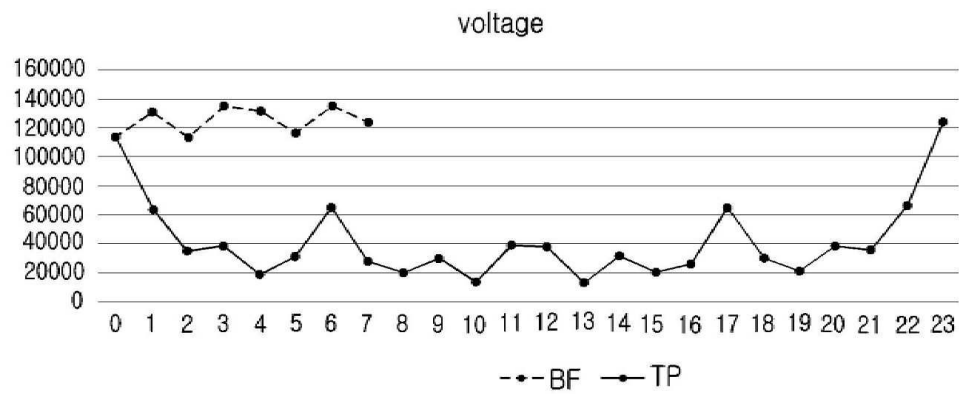
도면5



도면6



도면7



도면8

