



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2020-0075294
(43) 공개일자 2020년06월26일

(51) 국제특허분류(Int. Cl.)
G06F 16/903 (2019.01) G06F 7/02 (2006.01)
(52) CPC특허분류
G06F 16/90344 (2019.01)
G06F 7/02 (2013.01)
(21) 출원번호 10-2018-0163813
(22) 출원일자 2018년12월18일
심사청구일자 2018년12월18일

(71) 출원인
인하대학교 산학협력단
인천광역시 미추홀구 인하로 100(용현동, 인하대학교)
(72) 발명자
심정섭
인천광역시 남동구 남동대로 860, 120동 601호(간석동, 간석래미안자이아파트)
박정훈
경기도 수원시 팔달구 권광로 373, 112동 2205호(우만동, 월드메르디앙 아파트)
김영호
인천광역시 미추홀구 낙섬동로 6, 102동 301호(용현동, 한국아파트)
(74) 대리인
양성보

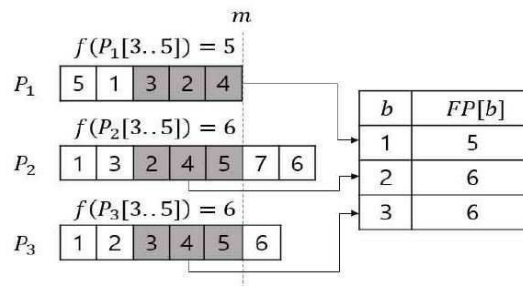
전체 청구항 수 : 총 6 항

(54) 발명의 명칭 공간효율적인 순위다중패턴매칭 알고리즘

(57) 요약

공간효율적인 순위다중패턴 매칭 알고리즘이 개시된다. 일 실시예에 따른 전처리부 및 검색부를 포함하는 해싱 장치에 의해 수행되는 공간효율적인 해싱 방법은, 상기 전처리부에서, 패턴집합의 마지막 q-그램에 대한 핑거프린트를 계산한 값을 핑거프린트 테이블에 저장하는 전처리 단계; 및 상기 검색부에서, 텍스트(T)의 검색위치(i)에 대한 핑거프린트를 계산한 값이 상기 핑거프린트 테이블(FP)에 존재하는지 여부를 판단하여 검색위치를 증가시키는 과정을 수행하는 검색 단계를 포함할 수 있다.

대 표 도 - 도7



(52) CPC특허분류

G06F 2207/025 (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호 1711069835

부처명 과학기술정보통신부

연구관리전문기관 한국연구재단

연구사업명 전략과제(산업수학)

연구과제명 [Ezbaro] 대용량 시계열 데이터의 반복성 검출과 패턴 예측을 위한 새로운 모델 및 효율적

인 알고리즘 개발

기 여 율 1/1

주관기관 인하대학교

연구기간 2018.03.01 ~ 2019.02.28

명세서

청구범위

청구항 1

전처리부 및 검색부를 포함하는 해싱 장치에 의해 수행되는 공간효율적인 해싱 방법에 있어서,
 상기 전처리부에서, 패턴집합의 마지막 q-그램에 대한 핑거프린트를 계산한 값을 핑거프린트 테이블에 저장하는 전처리 단계; 및
 상기 검색부에서, 텍스트(T)의 검색위치(i)에 대한 핑거프린트를 계산한 값이 상기 핑거프린트 테이블(FP)에 존재하는지 여부를 판단하여 검색위치를 증가시키는 과정을 수행하는 검색 단계
 를 포함하는 공간효율적인 해싱 방법.

청구항 2

제1항에 있어서,
 상기 공간효율적인 해싱 방법은,
 패턴집합에 존재하는 q-그램에 대한 핑거프린트 관련 정보를 저장하여 공간복잡도를 감소시키는
 공간효율적인 해싱 방법.

청구항 3

제1항에 있어서,
 상기 검색 단계는,
 상기 핑거프린트 테이블에 상기 텍스트의 검색위치에 대한 핑거프린트를 계산한 값이 존재하지 않을 경우, 순위 동형인 패턴이 없는 것으로 판단하고, 상기 검색위치를 1 증가시키는 단계
 를 포함하는 공간효율적인 해싱 방법.

청구항 4

제1항에 있어서,
 상기 전처리 단계는,
 상기 패턴집합이 주어졌을 때, 각 패턴의 마지막 q-그램에 대한 핑거프린트와 관련된 정보를 핑거프린트 테이블에 저장하는 단계
 를 포함하는 공간효율적인 해싱 방법.

청구항 5

제1항에 있어서,
 상기 전처리 단계는,
 상기 패턴집합에 대한 순위동형을 확인하기 위하여 위치테이블(POS)를 계산하는 단계
 를 포함하는 공간효율적인 해싱 방법.

청구항 6

해싱 장치에 있어서,
 패턴집합의 마지막 q-그램에 대한 핑거프린트를 계산한 값을 핑거프린트 테이블에 저장하는 전처리부; 및

텍스트(T)의 검색위치(i)에 대한 핑거프린트를 계산한 값이 상기 핑거프린트 테이블(FP)에 존재하는지 여부를 판단하여 검색위치를 증가시키는 과정을 수행하는 검색부를 포함하는 해싱 장치.

발명의 설명

기술 분야

[0001] 아래의 설명은 순위다중패턴매칭 문제에 대한 공간효율적인 해싱기반 알고리즘에 관한 것이다.

배경 기술

[0003] 길이가 같은 두 문자열 x, y 가 주어졌을 때, x 와 y 의 같은 위치에 있는 문자들의 순위가 모두 일치하면 x 와 y 는 순위동형(order-isomorphic)이라 한다. 예를 들어, 두 문자열 $A = (10, 20, 15, 30, 25)$ 와

$B = (7, 13, 10, 19, 16)$ 에서 각 문자들의 순위는 (1, 3, 2, 5, 4)로 동일하므로 A와 B는 순위동형이다. 순위패턴매칭문제(order-preserving pattern matching problem)는 길이가 각각 m 과 n 인 문자열 P 와 T 가 주어졌을 때, P 와 순위동형인 T 의 모든 부분문자열을 찾는 문제이다. 순위패턴매칭문제는 주가 변동분석, 음악 표절 분석 등과 같이 순위의 순서가 중요한 분야에 응용될 수 있다. 비특허문헌 1(J. Kim, P. Eades, R. Fleischer, S. H. Hong, C. S. Iliopoulos, K. Park, S. J. Puglisi, and T. Tokuyama, "Order-preserving matching," *Theoretical Computer Science*, Vol. 525, pp. 68-79, 2014.)과 비특허문헌 2(M. Kubica, T. Kulczynski, J. Radoszewski, W. Rytter, T. Walen, "A linear time algorithm for consecutive permutation pattern matching," *Information Processing Letters*, Vol. 113, No. 12, pp. 430-433, 2013.)에서는 Knuth-Morris-Pratt 알고리즘을 기반으로, 비특허문헌 3(S. Cho, J. C. Na, K. Park, J. S. Sim, "A fast algorithm for order-preserving pattern matching," *Information Processing Letters*, Vol. 115, No. 2, pp. 397-402, 2015.)에서는 Horspool 알고리즘을 기반으로 순위패턴매칭문제를 해결하였다.

[0004] 순위다중패턴매칭문제(order-preserving multiple pattern matching problem)는 k 개의 패턴들의 집합 $\hat{P} = \{P_1, P_2, \dots, P_k\}$ 과 길이가 n 인 텍스트 T 가 주어졌을 때, $\hat{P}$ 에 속한 $P_b (1 \leq b \leq k)$ 와 T 가 순위동형인 T 의 모든 부분문자열을 찾는 문제이다. 순위다중패턴매칭문제에 대한 최적 솔루션은 아직 알려지지 않았다. $\hat{P}$ 에 속한 패턴 중 가장 짧은 패턴의 길이를 m , 가장 긴 패턴의 길이를 $\overline{m}$, $\hat{P}$ 에 속한 패턴들의 길이의 총합을 M 이라 할 때, 순위다중패턴매칭문제를 비특허문헌 1에서는 Aho-Corasick 오토마타를 이용하여

$O(n \log M)$ 시간에 해결하였다. 비특허문헌 4(M. Han, M. Kang, S. Cho, G. Gu, J. S. Sim, K. Park, "Fast multiple order-preserving matching algorithms," *IWOCA*, Vol. 9538, pp. 248-259, Oct. 2015.)에서는 다중패턴매칭에서 평균수행시간이 빠른 알고리즘으로 잘 알려진 Wu-Manber 알고리즘을 순위다중패턴매칭문제에 이용한 알고리즘을 제시하였다. 이를 위하여 q-그램 기반 오문자규칙(bad-character heuristic)을 적용하여 전처리 단계는 $O(q! + kmq \log q + M \log \overline{m})$ 시간, 검색 단계는 평균 $O((n/m) \log M)$ 시간에 해결하였다. 그리고, Karp-Rabin 알고리즘을 이용하여 전처리 단계는 $O(m)$ 시간, 검색 단계는 $O(n)$ 시간에 해결하였다. 비특허문헌 5(Y. Shin, Y. Kim, J. S. Sim, "Parallel Computation of Z-function for Order-Preserving Multiple Pattern Matching," Proc. of the 43th KIISE Winter Conference, pp. 1812-1814, 2016. (in Korean)))에서는 순위다중패턴매칭을 위한 Z-함수를 $O(n)$ 시간에 계산하는 병렬알고리즘을 제시하

였다.

- [0005] 한편, q -그램 기반 Wu-Manber 알고리즘에서 사용되는 테이블은 $O(q!)$ 의 공간을 사용하며 q 의 크기가 증가할수록, 사용되지 않는 공간이 증가할 수 있다.

발명의 내용

해결하려는 과제

- [0007] 이동테이블과 해시테이블을 사용하지 않고 $O(k+M)$ 공간의 테이블을 이용하여 순위다중패턴매칭문제를 해결하는 방법을 제공할 수 있다.

과제의 해결 수단

- [0009] 전처리부 및 검색부를 포함하는 해싱 장치에 의해 수행되는 공간효율적인 해싱 방법은, 상기 전처리부에서, 패턴집합의 마지막 q -그램에 대한 핑거프린트를 계산한 값을 핑거프린트 테이블에 저장하는 전처리 단계; 및 상기 검색부에서, 텍스트(T)의 검색위치(i)에 대한 핑거프린트를 계산한 값이 상기 핑거프린트 테이블(FP)에 존재하는지 여부를 판단하여 검색위치를 증가시키는 과정을 수행하는 검색 단계를 포함할 수 있다.
- [0010] 상기 공간효율적인 해싱 방법은, 패턴집합에 존재하는 q -그램에 대한 핑거프린트 관련 정보를 저장하여 공간복잡도를 감소시킬 수 있다.
- [0011] 상기 검색 단계는, 상기 핑거프린트 테이블에 상기 텍스트의 검색위치에 대한 핑거프린트를 계산한 값이 존재하지 않을 경우, 순위동형인 패턴이 없는 것으로 판단하고, 상기 검색위치를 1 증가시키는 단계를 포함할 수 있다.
- [0012] 상기 전처리 단계는, 상기 패턴집합이 주어졌을 때, 각 패턴의 마지막 q -그램에 대한 핑거프린트와 관련된 정보를 핑거프린트 테이블에 저장하는 단계를 포함할 수 있다.
- [0013] 상기 전처리 단계는, 상기 패턴집합에 대한 순위동형을 확인하기 위하여 위치테이블(POS)를 계산하는 단계를 포함할 수 있다.
- [0014] 전처리부 및 검색부를 포함하는 해싱 장치는, 패턴집합의 마지막 q -그램에 대한 핑거프린트를 계산한 값을 핑거프린트 테이블에 저장하는 전처리부; 및 텍스트(T)의 검색위치(i)에 대한 핑거프린트를 계산한 값이 상기 핑거프린트 테이블(FP)에 존재하는지 여부를 판단하여 검색위치를 증가시키는 과정을 수행하는 검색부를 포함할 수 있다.

발명의 효과

- [0016] 순위다중패턴매칭문제를 해결하기 위하여 해싱기반 알고리즘의 공간복잡도와 전처리 단계의 시간복잡도를 개선할 수 있다.

도면의 간단한 설명

- [0018] 도 1은 일 실시예에 따른 해싱 장치의 구성을 설명하기 위한 블록도이다.
- 도 2는 일 실시예에 따른 해싱 장치의 공간효율적인 해싱 방법을 설명하기 위한 흐름도이다.
- 도 3 및 도 4는 일 실시예에 있어서, 다우존스지수 텍스트에 대한 OH와 FPA의 전처리 단계와 검색 단계의 수행 시간을 나타낸 그래프이다.

도 5는 및 도 6은 일 실시예에 있어서, 랜덤데이터 텍스트에 대한 OH와 FPA의 전처리 단계와 검색 단계 수행시간을 나타낸 그래프이다.

도 7은 일 실시예에 있어서, FP를 계산한 예이다.

발명을 실시하기 위한 구체적인 내용

이하, 실시예를 첨부한 도면을 참조하여 상세히 설명한다.

Wu-Manber 알고리즘 기반의 순위다중패턴매칭 알고리즘을 개선하여 공간복잡도를 $O(q! + k + M)$ 공간에서 $O(k + M)$ 공간으로 줄인 알고리즘을 제시한다. 다우존스지수 텍스트와 랜덤데이터 텍스트에 대한 실험 결과, 실시예에서 제안한 알고리즘은 q-그램 기반 Wu-Manber 알고리즘에 비해 전처리 시간은 빠르지만 검색시간을 느려져, 총 수행시간은 다소 느린 결과를 보였다.

실시예에서 설명하는 알고리즘에 대한 설명의 이해를 돕기 위하여 용어를 정의하기로 한다. 정수 문자집합 Σ 로 구성된 문자열 x 가 주어졌을 때, x 의 길이를 $|x|$, x 의 i 번째 문자를 $x[i]$ 로 표기한다. 문자열 x 의 i 번째부터 j 번째까지의 부분문자열을 $x[i..j]$ 로 표기한다. Σ 로 구성된 k 개의 패턴 집합을 $\hat{P} = \{P_1, P_2, \dots, P_k\}$ 라고 할 때, $\hat{P}$ 에 속한 패턴 중 가장 짧은 패턴의 길이를 m , 가장 긴 패턴의 길이

를 $\bar{m}$ 로 표기하고, $\hat{P}$ 에 속한 모든 패턴 길이의 합, 즉 $\sum_{i=1}^k |P_i|$ 을 M 으로 표기한다.

순위동형을 확인하기 위한 문자열 순위관계 표현방법으로 접두사표현법(prefix representation)이 있다. 문자열 x 에 대한 접두사표현법 $Pre(x)$ 는 다음과 같이 정의된다.

$$Pre(x)[i] = |\{j : x[j] \leq x[i] \ \forall 1 \leq j < i\}|$$

$Pre(x)$ 는 $x[1..i-1]$ 에서 $x[i]$ 보다 작거나 같은 문자의 개수를 저장한다. 예를 들면, $P=(35, 40, 23, 40, 40)$ 이 주어질 때, P 에 대한 접두사표현은 $Pre(P)=(0, 1, 0, 3, 4)$ 이다. 두 문자열 x 와 y 가 $|x|=|y|$ 이고, $Pre(x)=Pre(y)$ 이면, x 와 y 는 서로 순위동형이고 $x \approx y$ 로 표기한다.

순위패턴매칭에서 길이가 1인 문자열의 순위는 항상 0이기 때문에, 종래에는 길이 q 인 문자열을 하나의 단위로 취급하는 q-그램(q-gram)과 q-그램의 접두사표현법에 대한 핑거프린트(fingerprint) 함수를 이용하여 순위패턴 매칭문제를 해결하였다. q-그램 x 에 대한 핑거프린트 함수 $f(x)$ 는 다음과 같이 정의된다.

$$f(x) = \sum_{k=1}^q Pre(x)[k] \cdot (k-1)! + 1$$

- [0028] q -그램 x 에 대한 $f(x)$ 의 범위는 $1 \sim q!$ 이다. 이하, $f(x)$ 를 x 의 핑거프린트라고 기재하기로 한다. $f(x)$ 는 순위통계트리(order-statistics tree)를 이용하여 $O(q \log q)$ 시간에 계산할 수 있다.
- [0029] 도 1은 일 실시예에 따른 해싱 장치의 구성을 설명하기 위한 블록도이고, 도 2는 일 실시예에 따른 해싱 장치의 공간효율적인 해싱 방법을 설명하기 위한 흐름도이다.
- [0030] 실시예에서는 이동테이블과 해시테이블을 사용하지 않고 $O(k+M)$ 공간의 테이블을 이용하여 순위다중패턴매칭문제를 해결하는 방법을 설명하기로 한다. $\hat{P}'$ 에 존재하는 q -그램에 대한 핑거프린트 관련 정보들을 저장함으로써 공간복잡도를 감소시키는 것이다. 핑거프린트 관련 정보들을 저장하는 테이블을 핑거프린트 테이블(fingerprint table)로 부르고, 이하, FP 로 표기하기로 한다.
- [0031] 이때, $\hat{P}'$ 는 각 패턴 $P_b (1 \leq b \leq k)$ 의 길이 m 인 접두사 $P_b[1..m]$ 을 P'_b 라고 하고, $P'_1, P'_2, \dots, P'_k$ 으로 구성된 패턴집합을 의미할 수 있다.
- [0032] $\hat{P}'$ 이 주어졌을 때, $FP[b] (1 \leq b \leq k)$ 는 각 패턴집합 P'_b 의 마지막 q -그램에 대한 핑거프린트, 다시 말해서, $f(P'_b[m-q+1..m])$ 을 저장한다.
- [0033] 표 1을 참고하면, 전처리 단계(210)에서 전처리부(110)는 각 q -그램 $P'_b[m-q+1..m]$ 의 핑거프린트를 계산한 뒤 각 $FP[b]$ 에 값을 저장할 수 있다. FP 는 $O(k)$ 공간이 필요하며 $O(kq \log q)$ 시간에 계산할 수 있다. 도 7을 참고하면, $m=5, q=3$ 일 때 FP 를 계산한 예이다.
- [0034] 추가적으로 순위동형을 확인하기 위하여 위치테이블 POS 를 계산할 수 있다. 구체적으로, 위치테이블 POS 는 다음과 같이 계산될 수 있다. P_b 에 대해 $P_b[i]$ 의 순위를 저장한 배열을 $\pi(P_b)$ 라고 하자. $\pi(P_b)$ 는 P_b 를 정렬하여 $O(|P_b| \log |P_b|)$ 시간에 계산할 수 있다. P_b 에 대한 위치테이블 POS 는 각 문자들의 순위별 위치를 저장한 테이블이고, $\pi(P_b)[i] = j$ 이면, $POS[j] = i$ 이다. 위치테이블 POS 는 $\pi(P_b)$ 가 계산되어 있는 경우, $O(|P_b|)$ 시간에 계산할 수 있다. 위치테이블 POS 는 $\hat{P}$ 는 각 패턴에 대해 모두 계산하므로 $O(M)$ 공간을 사용하고, 최악의 경우 $O(M \log \bar{m})$ 시간에 계산할 수 있다. 이에 따라, 전처리 단계에서 $O(k+M)$ 의 공간을 사용하고, 총 $O(kq \log q + M \log \bar{m})$ 시간에 테이블들을 계산할 수 있다.

[0035] 표 1: FPA의 전처리 단계

FPA_Preprocessing	
Input : pattern set $\hat{P}$	
Output : position table POS , fingerprint table FP	
1	Compute POS
2	$m \leftarrow \min \{l : l = P_b , P_b \in \hat{P}\}$
3	$\hat{P}' \leftarrow \{P'_b P'_b = P_b[1..m], P_b \in \hat{P}\}$
4	$i \leftarrow 1, k \leftarrow \hat{P} $
5	while $i \leq k$ do
6	$x \leftarrow f(P'_i[m-q+1..m])$
7	$FP[i] \leftarrow x$
8	$i \leftarrow i+1$
9	end while

[0036]

[0037]

표 2를 참고하면, 검색 단계(220)에서 검색부(120)는 $T[i-q+1..i] (m \leq i \leq n)$ 의 핑거프린트를 계산하고, 핑거프린트 테이블(FP)에 해당하는 핑거프린트가 존재하지 않을 경우, 순위동형인 패턴이 존재하지 않는다는 뜻이므로, i 를 1 증가시키고 이러한 과정을 반복할 수 있다. $FP[b]$ 에 해당 핑거프린트가 존재하면, P_b 와 $T[i-m+1..(i-m+1)+|P_b|]$ 가 순위동형 여부를 위치테이블을 사용하여 확인할 수 있다. 최악의 경우, 모든 위치 i 에서 T 의 부분문자열과 모든 패턴들이 순위동형을 확인할 수 있다. 이때, 각 위치 i 마다 $O(M+q \log q)$ 시간이 필요하며, 검색 단계는 총 $O((M+q \log q)n)$ 시간에 수행할 수 있다.

[0038] 표 2: FPA의 검색 단계

FPA_Search
Input : pattern set $\hat{P}$, text T , position table POS , fingerprint table FP
Output : Occurred pattern numbers and their positions in T
1 $n \leftarrow T $
2 $m \leftarrow \min \{l : l = P_b , P_b \in \hat{P}\}$
3 $i \leftarrow m, k \leftarrow \hat{P} $
4 while $i \leq n$ do
5 $x \leftarrow f(T[i-q+1..i])$
6 for $j=1$ to k
7 if $x = FP[j]$ then
9 if $P_j \approx T[i-m+1..(i-m+1)+ P_j]$ then
10 print "Pattern j Matched at i "
11 end if
12 end if
13 end for
14 $i \leftarrow i+1$
15 end while

[0039]

[0040]

도 3 및 도 4는 일 실시예에 있어서, 다우존스지수 텍스트에 대한 OH와 FPA의 전처리 단계와 검색 단계의 수행 시간을 나타낸 그래프이다.

[0041]

도 3을 참고하면, 도3(a)는 $m=200$ 일 때, k 변화에 따른 전처리 시간, 도 3(b)는 $m=200$ 일 때, k 변화에 따른 검색 시간을 나타낸 그래프이다. 도 4를 참고하면, 도 4(a)는 $k=2000$ 일 때, m 변화에 따른 전처리 시간, 도 4(b)는 $k=2000$ 일 때, m 변화에 따른 검색 시간을 나타낸 그래프이다.

[0042]

예를 들면, CPU는 Intel Core i7-6700, RAM은 32GB, OS는 Windows 10 (64bit), 개발 툴은 Visual Studio 2015, 프로그래밍 언어는 C++를 사용한 실험환경을 통하여 실험이 수행될 수 있다. 실험데이터는 rand() 함수를 이용하여 생성한 랜덤데이터 텍스트와 일별마감기준 다우존스지수(DJA)를 텍스트로 사용할 수 있다. 랜덤데이터 텍

스트의 길이는 50000, 다우존스지수 텍스트의 길이는 34685로 생성하였고, 패턴집합 $\hat{P}$ 은 T 의 임의의 위치들에서 추출되어 생성될 수 있다. q 의 길이는 6-9로 사용하고, 패턴개수 k 는 200부터 2000까지 20씩 증가시키고 패턴길이 m 은 20부터 200까지 20씩 증가시키며 실험을 진행할 수 있다. 종래의 Wu-Manber 알고리즘을 기반의 순위다중패턴매칭 알고리즘을 OH, 실시예에서 제안된 알고리즘을 FPA로 기재하기로 한다.

[0043]

도 3 및 도 4를 참고하면, 다우존스지수 텍스트에 대하여 k 와 m 을 변화시키며 OH와 FPA의 수행시간을 비교한 것이다. 도 3(a)와 도 3(b)는 $m=200$ 으로 고정시키고, k 를 200부터 2000까지 20씩 증가시키며 전처리 단계와 검색 단계의 수행시간을 비교한 결과이다. 도 4(a)와 도 4(b)는 $k=2000$ 으로 고정시키고 m 을 20부터 200까지 20씩 증가시키며 전처리 단계와 검색 단계의 수행시간을 비교한 결과이다. 실험결과, OH의 총 수행시간은 $q=7$ 일 때 가장 빠르다. $q=7$, $m=200$, $k=2000$ 일 때, 전처리 단계의 수행시간은 OH가 71ms, FPA가 32ms로서 FPA가 OH보다 약 2.2배 빠르게 수행되었으며, 검색 단계는 OH가 5ms, FPA가 43ms로서 OH가 FPA보다 9.6배 빠르게 수행되었다. 총 수행시간은 OH가 76ms, FPA가 75ms로서 유사한 결과를 보였다.

[0044]

도 5는 및 도 6은 일 실시예에 있어서, 랜덤데이터 텍스트에 대한 OH와 FPA의 전처리 단계와 검색 단계 수행 시간을 나타낸 그래프이다.

[0045]

도 5 및 도 6은 랜덤데이터 텍스트에 대하여 k 와 m 을 변화시키며 OH와 FPA의 수행시간을 비교한 것이다. 도 5(a)와 도 5(b)는 $m=200$ 으로 고정시키고, k 를 200부터 2000까지 20씩 증가시키며 전처리 단계와 검색 단계의 수행시간을 비교한 결과이다. 도 6(a)와 도 6(b)는 $k=2000$ 으로 고정시키고, m 을 20부터 200까지 20씩 증가시키며 전처리 단계와 검색 단계의 수행시간을 비교한 결과이다. 실험결과, OH의 총 수행시간은 $q=7$ 일 때 가장

빠르다. $q=7$, $m=200$, $k=2000$ 일 때, 전처리 단계의 수행시간은 OH가 71ms, FPA가 32ms로서 FPA가 OH보다 약 2.21배 빠르게 수행되었으며, 검색 단계는 OH가 5ms, FPA가 61ms로서 OH가 FPA보다 12.2배 빠르게 수행되었다. 총 수행시간은 OH가 76ms, FPA가 96ms로서 OH가 약 1.22배 빠르게 수행되었다.

[0046] $k=200$, $m=20$ 일 때 FPA는 q 의 크기에 상관없이 328KB의 공간이 사용되었다. 반면, OH는 q 의 크기가 증가할수록 테이블들에 사용되는 공간 사용량이 증가하는 모습을 보였다. 다시 말해서, $q=6$ 일 때, 약 335KB의 공간을 사용하였고, $q=9$ 일 때 약 3173KB의 공간을 사용하여 FPA의 테이블보다 약 10배 더 많은 크기를 사용하였다.

[0047] 실시예에 따르면, 순위다중패턴매칭문제를 해결하기 위하여 해싱기반 알고리즘의 공간복잡도와 전처리 단계의 시간복잡도를 개선할 수 있다. OH는 k 와 m 이 증가할수록 이동테이블 계산시간이 증가하기 때문에 전처리 단계에서는 FPA가 OH보다 빠른 수행시간을 보임을 확인할 수 있다.

[0048] 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), 마이크로컴퓨터, 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 콘트롤러를 포함할 수 있다. 또한, 병렬 프로세서(parallel processor)와 같은, 다른 처리 구성(processing configuration)도 가능하다.

[0049] 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, 가상장치(virtual equipment), 컴퓨터 저장 매체 또는 장치에 구체화(embody)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.

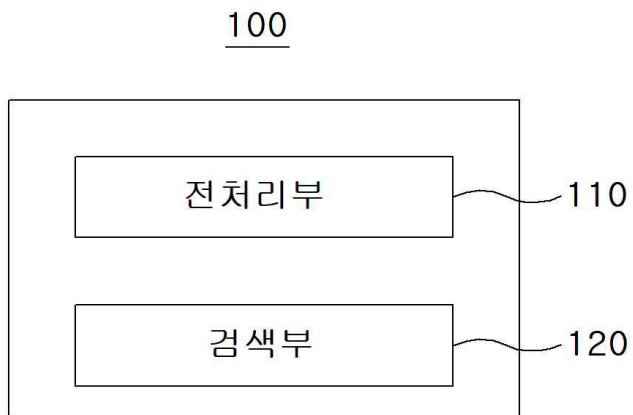
[0050] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 실시예를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic 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.

[0051] 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.

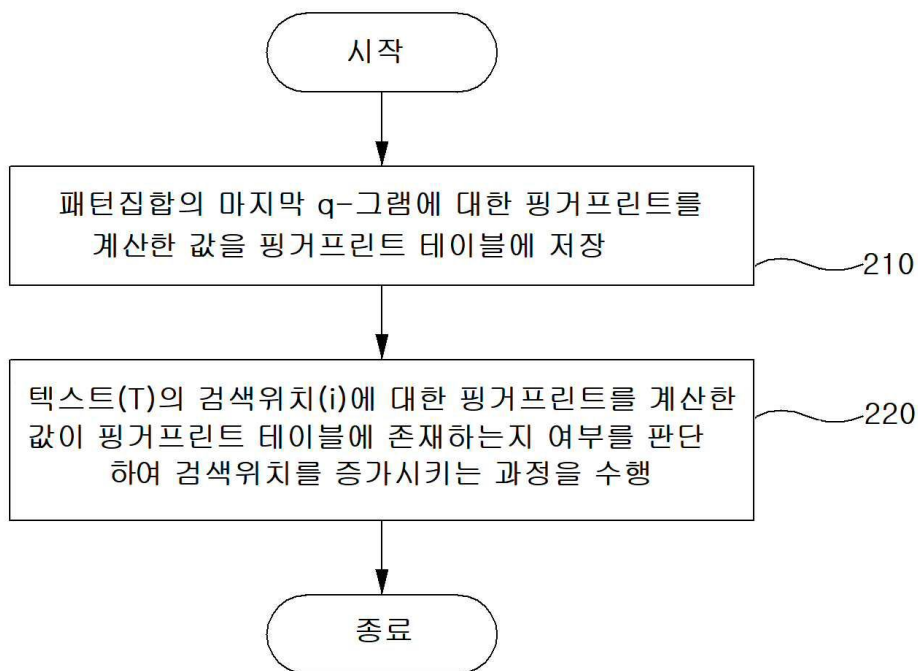
[0052] 그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.

도면

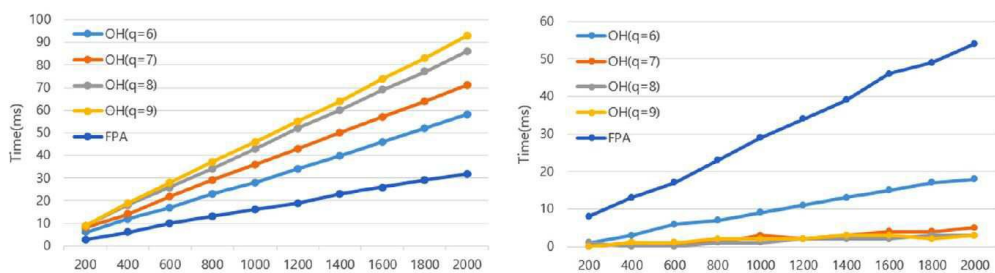
도면1



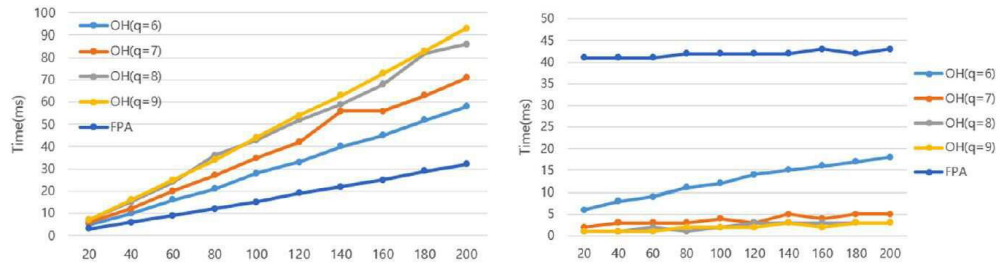
도면2



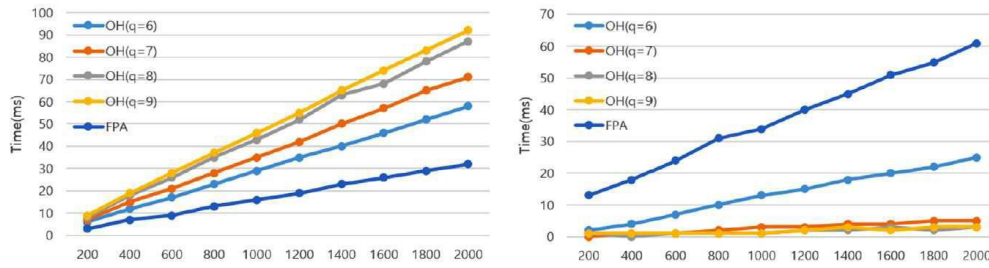
도면3



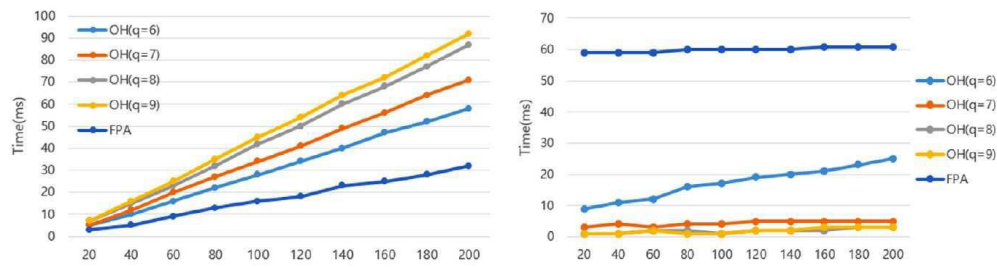
도면4



도면5



도면6



도면7

