



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2016년10월13일
(11) 등록번호 10-1666307
(24) 등록일자 2016년10월07일

(51) 국제특허분류(Int. Cl.)
G06F 7/22 (2006.01) G06F 12/06 (2006.01)
G06F 12/08 (2016.01)
(52) CPC특허분류
G06F 7/22 (2013.01)
G06F 12/063 (2013.01)
(21) 출원번호 10-2015-0067407
(22) 출원일자 2015년05월14일
심사청구일자 2015년05월14일
(56) 선행기술조사문헌
KR1020010041803 A*
KR1020050060553 A*
*는 심사관에 의하여 인용된 문헌
기술이전 희망 : 기술양도

(73) 특허권자
영남대학교 산학협력단
경상북도 경산시 대학로 280 (대동)
군산대학교산학협력단
전라북도 군산시 대학로 558 (미룡동, 군산대학교)
(72) 발명자
최규상
대구광역시 수성구 신매로 51, 229동 1309호 (신매동, 시지천마타운)
은병원
전라북도 군산시 미룡로 41, 201동 401호 (미룡동, 미룡주공2차아파트)
이인규
인천광역시 남구 석정로 291, 다동 304호 (도화동, 태화아파트)
(74) 대리인
김중선, 이형석

전체 청구항 수 : 총 5 항

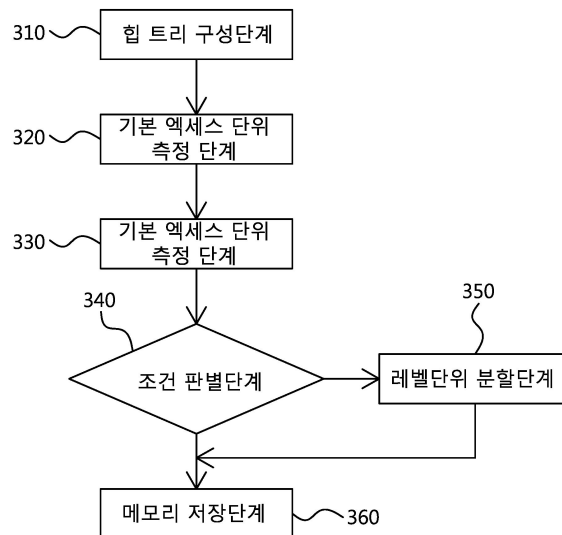
심사관 : 지정훈

(54) 발명의 명칭 메모리 장치 기반의 힙 정렬 방법 및 장치

(57) 요약

본 발명은 배열 기반의 힙 정렬 방법 및 장치에 관한 것으로서 이진트리의 데이터들을 서브 트리 단위를 기준으로 외부 메모리 장치의 기본 액세스 단위에 저장함으로써 힙 정렬 수행 시 외부 메모리에 대한 접속(I/O) 빈도를 낮출 수 있춰 힙 정렬 속도를 향상시킬 수 있다.

대표도 - 도3



(52) CPC특허분류

G06F 12/0802 (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호 1345208508
 부처명 교육부
 연구관리전문기관 한국연구재단
 연구사업명 일반연구자지원(교육부)
 연구과제명 MRAM 기반의 임베디드 운영체제 연구
 기여율 1/3
 주관기관 영남대학교
 연구기간 2013.11.01 ~ 2016.10.31

이 발명을 지원한 국가연구개발사업

과제고유번호 1415132504
 부처명 산업통상자원부
 연구관리전문기관 한국에너지기술평가원
 연구사업명 에너지자원융합원천기술개발
 연구과제명 ICT 기술융합을 기반으로 한 국민발전소 비즈니스 플랫폼 개발
 기여율 1/3
 주관기관 (재)차세대융합기술연구원
 연구기간 2013.08.01 ~ 2014.07.31

이 발명을 지원한 국가연구개발사업

과제고유번호 1345204589
 부처명 미래창조과학부
 연구관리전문기관 한국연구재단
 연구사업명 일반연구자지원(미래부)
 연구과제명 차세대 비휘발성 메모리 기반의 빅 데이터 처리 알고리즘 개발
 기여율 1/3
 주관기관 재단법인 차세대융합기술연구원
 연구기간 2013.06.01 ~ 2016.05.31

명세서

청구범위

청구항 1

데이터들을 이진트리로 배열하는 힙 트리 구성단계;
 상기 힙 트리를 일정한 크기의 서브트리로 분할하는 서브트리 분할단계; 및
 상기 데이터들을 상기 서브트리를 기준 단위로 메모리 장치에 저장하는 메모리 저장단계를 포함하고,
 상기 메모리 장치의 기본 액세스 단위의 크기를 측정하는 측정단계를 더 포함하는 힙 정렬 방법.

청구항 2

삭제

청구항 3

제1항에 있어서,
 상기 서브트리의 크기는 상기 메모리 장치의 기본 액세스 단위의 크기와 동일한 힙 정렬 방법.

청구항 4

제1항에 있어서,
 상기 서브트리 중 최하단의 서브트리를 구성하는 데이터의 크기가 판별기준을 충족하는지 여부를 판별하는 판별 단계; 및
 상기 최하단의 서브트리를 구성하는 데이터의 크기가 상기 기본 액세스 단위의 크기보다 작으면 최하단의 서브 트리에 속하는 데이터들을 레벨 단위로 분할하는 레벨단위 분할단계를 더 포함하는 힙 정렬 방법.

청구항 5

제4항에 있어서,
 상기 판별기준은 상기 기본 액세스 단위의 크기를 충족하는지 여부를 판별하는 힙 정렬 방법.

청구항 6

데이터들을 이진트리로 이루어진 힙 트리 구성하는 힙 트리 구성부;
 메모리 장치의 기본 액세스 단위의 크기를 측정하는 측정부;
 상기 이진트리를 기본 액세스 단위의 크기를 갖는 서브트리로 분할하는 서브트리 분할부;
 최하단 서브트리에 저장된 데이터의 크기가 소정의 판별기준을 충족하는지 여부를 판별하는 판별부; 및
 상기 판별부에서 최하단 서브트리가 상기 판별기준을 충족하지 못하면 최하단 서브트리에 저장된 데이터들을 레벨단위로 분할하는 레벨단위 분할부를 포함하며,
 상기 판별기준은 상기 기본 액세스 단위의 크기를 충족하는지 여부인 것을 특징으로 하는 힙 정렬 장치.

청구항 7

삭제

발명의 설명

기술 분야

[0001] 본 발명은 배열 기반의 힙 정렬 방법에 관한 것으로서, 메모리의 I/O 액세스를 감소시켜 힙 정렬 성능을 향상시키는 데이터 정렬 방법 및 장치에 관한 것이다.

배경 기술

[0002] 힙 정렬(Heap Sort)은 힙(Heap)이라는 이진트리(Binary Tree)로 이루어진 힙(Heap)이라는 자료구조를 사용하는 정렬방식으로서 각 노드들의 데이터를 비교하여 데이터를 정렬한다. 힙 정렬은 루트노드를 최대값으로 하여 부모노드가 자식노드보다 큰 값을 갖는 최대 힙(Maximum Heap) 또는 반대로 루트노드를 최소값으로 하여 부모노드가 자식노드보다 작은 값을 갖는 최소 힙(minimum Heap)을 구성한 후 부모노드와 자식노드의 값을 비교하면서 데이터를 정렬한다.

[0003] 도1은 외부 메모리(110)와 메인메모리(120) 사이의 접속(I/O)을 개략적으로 나타내는 도면이다.

[0004] 도1에 도시된 바와 같이, 메모리 장치(110, 120)는 형태에 따라 기본 액세스 단위(140)가 있으며, 컴퓨터가 메모리 장치(110, 120)에 접속하여 데이터를 읽고 쓸 때에는 기본 액세스 단위(140)를 기준으로 동작이 이루어진다.

[0005] 힙 정렬 시 정렬 대상이 되는 전체 데이터의 크기가 크지 않은 경우에는 외부 메모리 장치(110)에 접속(I/O, Input/Output)하여 모든 데이터들을 컴퓨터의 메인메모리(120)로 전송할 수 있지만, 전체 데이터의 규모가 큰 경우 데이터 전체를 메인메모리(120)로 전송할 수 없으므로 힙 정렬을 위한 연산을 수행할 때마다 외부 메모리 장치(110)에 접속하여 데이터를 전송해야 한다.

[0006] 이 때, 외부 메모리 장치(110)에 접속하는 속도는 상대적으로 느리기 때문에 외부 메모리 장치(110)에 접속하는 빈도가 높을수록 힙 정렬 수행속도가 느려지게 된다.

[0007] 도2는 종래 기술에 의한 힙 정렬에서의 데이터 저장 방식을 나타내는 도면이다.

[0008] 도2(a)에 도시된 바와 같이, 종래 기술에 의한 힙 정렬에서는 이진트리(200)로 배열되는 데이터들이 메모리(110)의 기본 액세스 단위(140)를 기준으로 루트 노드(210)부터 차순위 레벨 순으로 저장된다.

[0009] 도2(b)에 도시된 바와 같이, 기본 액세스 단위(140)의 크기가 1024KB인 경우 종래 기술에 의한 힙 정렬은 루트 노드(210)를 기점으로 1024KB를 충족하는 레벨까지 외부 메모리 장치(110)에 저장되고, 다시 차순위 레벨부터 1024KB를 충족하는 레벨까지의 데이터가 외부 메모리 장치(110)에 저장된다.

[0010] 힙 정렬은 부모노드와 자식노드를 비교하며 정렬을 수행하는 반면에, 종래 기술에 의한 힙 정렬 데이터 저장 방식은 부모 노드와 자식 노드의 데이터가 서로 다른 기본 액세스 단위(140)에 저장되므로 외부 메모리 장치(110)에 대한 접속(I/O) 빈도가 높아진다. 외부 메모리 장치(110)에 대한 접속(I/O) 빈도가 높아지면 데이터 전송에 많은 시간을 소비하게 되므로 전체적인 힙 정렬 수행속도가 느려지는 문제점이 있다.

[0011] 특히, 빅데이터와 같이 대규모 데이터를 처리하는 시스템에 있어서 힙 정렬 수행속도가 느려지는 것은 데이터 처리에 대한 큰 문제가 될 수 있다.

발명의 내용

해결하려는 과제

[0012] 본 발명의 목적은 힙 정렬 수행 시 외부 메모리 접속(I/O) 빈도를 감소시킬 수 있는 메모리 장치 기반의 힙 정렬 방법 및 장치를 제공하는 것이다.

과제의 해결 수단

[0013] 본 발명에 의한 힙 정렬 방법은 데이터들을 이진트리로 배열하는 힙 트리 구성단계, 상기 힙 트리를 일정한 크기의 서브트리로 분할하는 서브트리 분할단계; 및 상기 데이터들을 상기 서브트리를 기준 단위로 메모리 장치에 저장하는 메모리 저장단계를 포함한다.

[0014] 또한, 상기 메모리 장치의 기본 액세스 단위의 크기를 측정하는 측정단계를 더 포함할 수 있다.

[0015] 또한, 상기 서브트리의 크기는 상기 메모리 장치의 기본 액세스 단위의 크기와 동일할 수 있다.

[0016] 여기에서, 상기 메모리 장치는 플래시 메모리 또는 하드디스크를 포함하는 외부 메모리 장치일 수 있다.

[0017] 또한, 상기 서브트리 분할단계는 루트 노드를 기준으로 하위의 서브트리로 분할하는 것을 특징으로 한다.

[0018] 또한, 상기 서브트리 중 최하단의 서브트리를 구성하는 데이터의 크기가 상기 기본 액세스 단위의 크기를 충족하는지 여부를 판별하는 판별단계를 더 포함할 수 있다.

[0019] 또한, 상기 최하단의 서브트리를 구성하는 데이터의 크기가 상기 기본 액세스 단위의 크기보다 작으면 최하단의 서브트리에 속하는 데이터들을 레벨 단위로 분할하는 레벨단위 분할단계를 더 포함할 수 있다.

발명의 효과

[0020] 본 발명에 따른 메모리 장치 기반의 힙 정렬 방법 및 장치는 이진트리의 데이터들을 서브 트리 단위를 기준으로 외부 메모리 장치의 기본 액세스 단위에 저장함으로써 힙 정렬 수행 시 외부 메모리에 대한 접속(I/O) 빈도를 낮춰 힙 정렬 수행속도를 향상시킬 수 있다.

도면의 간단한 설명

[0021] 도1은 종래 기술에 의한 힙 정렬에서의 데이터 저장 방식을 나타내는 도면이다.

도2는 메인메모리와 외부 메모리 사이의 접속을 개략적으로 나타내는 도면이다.

도3은 본 발명의 실시예에 따른 힙 정렬 방법의 순서를 나타내는 순서도이다.

도4는 본 발명의 실시예에 따라 서브트리로 분할된 이진트리를 개략적으로 나타내는 도면이다.

도5는 최하단 서브트리의 데이터 크기가 기본 액세스 단위를 충족하지 못하는 경우의 이진트리를 나타내는 도면이다.

도6은 최하단 서브트리 구조를 레벨단위로 변경한 형태의 이진트리를 나타내는 도면이다.

도7은 본 발명의 실시예에 따른 힙 정렬 장치를 나타내는 도면이다.

발명을 실시하기 위한 구체적인 내용

[0022] 아래에서는 첨부한 도면을 참고로 하여 본 발명의 실시예에 대하여 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 상세히 설명한다. 그러나 본 발명은 여러 가지 상이한 형태로 구현될 수 있으며 여기에서 설명하는 실시예에 한정되지 않는다. 그리고 도면에서 본 발명을 명확하게 설명하기 위해서 설명과 관계없는 부분은 생략하였으며, 명세서 전체를 통하여 유사한 부분에 대해서는 유사한 도면 부호를 붙였다.

[0023] 도3은 본 발명의 실시예에 따른 힙 정렬 방법(300)의 순서를 나타내는 순서도이다.

[0024] 도3에 도시된 바와 같이, 본 발명의 실시예에 따른 힙 정렬 방법(300)은 정렬 대상이 되는 데이터들을 대상으로 이진트리(200)로 이루어진 힙 트리를 구성하는 힙 트리 구성단계(310), 외부 메모리 장치(110)의 기본 액세스 단위(140)의 크기를 측정하는 측정단계(320); 이진트리(200)를 기본 액세스 단위(140)의 크기를 기준으로 루트노드(Root node)부터 최하단 노드(Leaf node)까지 서브트리(220)로 분할하는 서브트리 분할단계(330), 최하단 서브트리(220)에 저장된 데이터의 크기가 기본 액세스 단위(140)의 크기를 충족하는지 여부를 판별하는 판별단계(340), 판별단계(340)에서 최하단 서브트리(220)가 상기 조건을 충족하지 못하면 최하단 서브트리(220)에 저장된 데이터들을 레벨단위(230)로 분할하는 레벨단위 분할단계(350) 및 상기 서브트리(220)와 레벨단위(230)로 분할된 데이터들을 기본 액세스 단위(140)로 외부 메모리 장치(110)에 저장하는 메모리 저장단계(360)를 포함한다.

[0025] 힙 트리 구성단계(310)는 이진트리(200)로 데이터들을 배열하는 단계로서, 정렬 대상이 되는 데이터들을 루트노드(210)를 기점으로 하위의 2개의 자식노드를 갖고, 상기 자식노드들은 부모노드가 되어 각 2개의 자식노드를 갖는 방식으로 트리를 구성된다. 이 때, 루트노드를 최대값으로 설정하고, 부모노드의 값이 자식노드의 값보다 크게 구성된 이진트리(200)를 최대힙이라 하고, 반대로 루트노드를 최소값으로 설정하고, 부모노드의 값이 자식노드의 값보다 작게 구성된 이진트리(200)를 최소힙이라 한다.

[0026] 기본 액세스 단위 측정단계(320)는 외부 메모리 장치(110)를 구성하는 기본 액세스 단위(140)의 크기를 측정하는 단계이다. 기본 액세스 단위(140)는 후술하는 서브트리 분할단계(330)에서 서브트리(220)의 크기를 설정하는 기준에 해당된다. 데이터들을 기본 액세스 단위(140)의 크기로 분할하여 저장함으로써 외부 메모리 장치(110)에 저장되는 데이터들의 크기를 최적화할 수 있고, 힙 정렬 수행 시 외부 메모리 장치(110)에 접속(I/O)할 때 전송하는 데이터량을 최대화할 수 있다.

[0027] 도3에서는 서브트리(220)의 크기를 설정하기 위한 기준으로서 기본 액세스 단위 측정단계(320)를 포함하는 힙 정렬 방법(300)이 도시되어 있지만, 서브트리(220)의 크기가 반드시 기본 액세스 단위(140)로 한정되어야 하는 것은 아니며, 힙 정렬을 수행하는 컴퓨팅 환경에 따라 다양하게 설정할 수도 있다.

[0028] 예를 들어, NAND 플래시 메모리는 페이지 단위로 읽기동작과 쓰기동작이 수행되고, 소거동작은 블록을 단위로 수행되는데, 외부 메모리 장치(110)가 NAND 플래시 메모리인 경우에는 페이지가 기본 액세스 단위(140)가 될 수 있다. 기본 액세스 단위(140)는 반드시 NAND 플래시 메모리의 페이지로 한정되는 것은 아니며, 외부 메모리 장

치(110)의 종류 및 형태에 따라 달라질 수 있다.

- [0029] 서브트리 분할단계(330)는 기본 액세스 단위 측정단계(320)에서 측정된 기본 액세스 단위(140)의 크기에 따라 서브트리(220)의 크기를 설정하고, 루트노드(210)를 기점으로 이진트리(200)를 구성하는 데이터들의 저장단위를 서브트리(220)로 분할하는 단계이다. 서브트리 분할단계(330)에 대해서는 도4를 참조하여 구체적인 설명을 하기로 한다.
- [0030] 도4는 본 발명의 실시예에 따라 서브트리(220)로 분할된 이진트리(200)를 개략적으로 나타내는 도면이다.
- [0031] 도4(a)에 도시된 바와 같이, 기본 액세스 단위(140)의 크기가 3개의 노드 값을 저장할 수 있는 크기라고 가정한다면, 분할되는 영역은 1개의 부모노드와 2개의 자식노드를 가진 서브트리(220)로 설정될 수 있다. 분할된 서브트리(220)는 루트노드(210)를 기점으로 최상위에 서브트리(220)가 구성되고, 다시 서브트리(220) 최하위 노드들을 기점으로 서브트리(220)가 구성된다.
- [0032] 도4(b)에 도시된 바와 같이, 기본 액세스 단위(140)의 크기가 1024KB인 경우 루트노드(210)를 기점으로 한 최상위의 서브트리(220)가 구성되고, 최상위 서브트리(220)에 포함된 최하위 노드들을 기점으로 한 서브트리(220)들이 구성된다. 이 때, 상위 서브트리(220)의 하위단에 연결될 수 있는 하위 서브트리(220)의 최대 수는 상위 서브트리(220)의 최하위 노드의 수에 의해 결정된다. 즉, 상위 서브트리(220)를 구성하는 최하위 노드의 수가 256개이면 최대 256개까지의 서브트리(220)가 최하위 노드에 연결될 수 있다.
- [0033] 상술한 바와 같이, 서브트리(220)의 크기는 외부 메모리 장치(110)의 기본 액세스 단위(140)의 크기를 기준으로 설정될 수 있지만, 반드시 이에 제한되어야 하는 것은 아니며, 메인메모리(120) 크기를 고려하여 설정되는 등 다양하게 설정될 수 있다.
- [0034] 판별단계(340)는 최하단 서브트리(220)를 구성하는 데이터의 크기가 외부 메모리 장치(110)의 기본 액세스 단위(140) 크기를 충족하는지 여부를 판별하는 단계이다. 판별단계(340)에 대한 구체적인 설명은 도5를 참조하여 설명하기로 한다.
- [0035] 도5는 최하단 서브트리(220)의 데이터 크기가 기본 액세스 단위(140)를 충족하지 못하는 경우의 이진트리(200)를 나타내는 도면이다.
- [0036] 도5에 도시된 바와 같이, 기본 액세스 단위(140)의 크기가 1024KB라고 가정하는 경우 루트노드(210)를 기점으로 한 최상위 서브트리(220)와 하위의 서브트리(220)는 데이터의 크기가 기본 액세스 단위(140)의 크기를 충족하지만, 최하단 서브트리(220)의 데이터 크기는 32KB에 불과하여 1024KB인 기본 액세스 단위(140)의 크기를 충족하지 못한다. 따라서, 판별단계(340)는 최하단 서브트리(220)의 데이터 크기가 기본 액세스 단위(140)의 크기를 충족하는지 여부를 판별한 후, 판별 결과에 따라 레벨단위 분할단계(350)의 수행여부를 결정한다.
- [0037] 레벨단위 분할단계(350)는 판별단계(340)의 결과에 따라 최하단 서브트리(220)의 데이터 크기가 기본 액세스 단위(140)의 크기를 충족하지 못하는 경우 저장단위를 서브트리(220) 구조에서 레벨단위(230) 구조로 변경한다. 레벨단위 분할단계(350)의 구체적인 설명은 도5 및 도6을 참조하여 설명하기로 한다.
- [0038] 도5에 도시된 바와 같이, 최하단 서브트리(220)의 데이터 크기는 기본 액세스 단위(140)의 크기를 충족하지 못

하므로 최하단 서브트리(220)들은 레벨단위 분할단계(350)에서 서브트리(220) 구조가 해제된다. 서브트리(220) 구조가 해제된 상태의 데이터들은 다시 레벨단위(230)로 저장단위의 구조가 변경된다.

[0039] 도6은 최하단 서브트리(220) 구조를 레벨단위(230)로 변경한 형태의 이진트리(200)를 나타내는 도면이다.

[0040] 도6에 도시된 바와 같이, 루트노드(210)를 기점으로 하위의 서브트리(220)가 구성되어 있지만, 이진트리(200)의 최하단에는 서브트리(220) 구조가 아닌 레벨단위(230)로 구성된다. 이와 같이, 최하단 서브트리(220)의 데이터 크기가 기본 액세스 단위(140)의 크기를 충족하지 못하는 경우 서브트리(220)를 구조로 저장단위를 구성하면 외부 메모리 장치(110)의 기본 액세스 단위(140) 사용률이 높아져 효율성을 떨어뜨린다. 예를 들어, 1024KB를 저장할 수 있는 기본 액세스 단위(140)에 32KB만을 저장하고, 사용하는 기본 액세스 단위(140)의 수는 많아지므로 효율이 떨어진다. 반면에, 최하단 서브트리(220)을 레벨단위(230)로 저장단위 구조를 변경하면 기본 액세스 단위(140)의 크기를 충족하는 저장단위로 사용할 수 있고, 보다 적은 기본 액세스 단위(140)의 사용이 가능해진다.

[0041] 도3에는 최하단 서브트리(220)의 데이터 크기가 기본 액세스 단위(140)의 크기를 충족하는지 여부만을 판별하는 방식이 도시되어 있지만, 이와 달리 최하단 서브트리(220) 데이터 크기를 일정한 충족률을 조건으로 판별하는 방식을 사용할 수도 있다. 예를 들어, 최하단 서브트리(220)의 데이터 크기가 기본 액세스 단위(140) 크기의 75%를 충족하는지 여부를 기준으로 판별할 수 있으며, 이 때, 기본 액세스 단위(140)의 크기가 1024KB인 경우 판별단계(340)의 판별기준은 768KB로 설정될 수 있다.

[0042] 즉, 판별기준이 높을수록 외부 메모리 장치(110)의 효율성을 높일 수 있는 반면에 부모노드와 자식노드가 서로 다른 기본 액세스 단위(140)에 저장될 가능성이 높아져 힙 정렬 수행속도가 느려질 수 있고, 반대로 판별기준이 낮을수록 외부 메모리 장치(110)의 효율성을 높일 수 있으나 힙 정렬 수행속도가 향상될 수 있다.

[0043] 메모리 저장단계(360)는 이진트리(200)를 구성하는 데이터들을 서브트리(220) 및 레벨단위(230)를 기준으로 외부 메모리 장치(110)에 저장하는 단계이다. 이 때, 서브트리(220) 및 레벨단위(230)의 크기가 기본 액세스 단위(140)의 크기로 설정된 경우, 서브트리(220) 및 레벨단위(230)는 각 1개의 기본 액세스 단위(140)들에 저장될 수 있다.

[0044] 본 발명의 실시예에 따른 힙 정렬 방법으로 데이터들이 저장되는 경우, 서브트리(220)를 기준으로 1개의 기본 액세스 단위(140)에 저장되므로 부모노드와 자식노드가 1개의 기본 액세스 단위(140)에 저장될 가능성이 높다. 따라서, 힙 정렬을 수행하는 동안 부모노드와 자식노드의 데이터를 비교하기 위해 루트노드를 기점으로 최하단 노드까지 1개의 경로를 따라 연결된 서브트리(220)만을 메인메모리(120) 전송하면 충분하고, 그만큼 외부 메모리 장치(110)에 접속해야하는 빈도가 줄어들어 힙 정렬 수행속도가 향상된다.

[0045] 도7은 본 발명의 실시예에 따른 힙 정렬 장치(400)를 나타내는 도면이다.

[0046] 도7에 도시된 바와 같이, 본 발명의 실시예에 따른 힙 정렬 장치(400)는 정렬 대상이 되는 데이터들을 대상으로 이진트리(200)로 이루어진 힙 트리를 구성하는 힙 트리 구성부(410), 외부 메모리 장치(110)의 기본 액세스 단위(140)의 크기를 측정하는 측정부(420), 이진트리(200)를 기본 액세스 단위(140)의 크기를 기준으로 루트 노드(Root node)부터 최하단 노드(Leaf node)까지 서브트리(220)로 분할하는 서브트리 분할부(430), 최하단 서브트리(220)에 저장된 데이터의 크기가 기본 액세스 단위(140)의 크기를 충족하는지 여부를 판별하는 판별부(440), 및 판별부(440)에서 최하단 서브트리(220)가 상기 조건을 충족하지 못하면 최하단 서브트리(220)에 저장된 데이터들을 레벨단위(230)로 분할하는 레벨단위 분할부(350)를 포함한다.

[0047] 본 발명의 실시예에 따른 힙 정렬 장치(400)는 도3 내지 도6에서 상술한 바와 같이 본 발명의 실시예에 따른 힙 정렬 방법을 수행함으로써 외부 메모리 장치(110)에 대한 접속 빈도를 낮춰 힙 정렬 수행 속도를 향상시킬 수 있다.

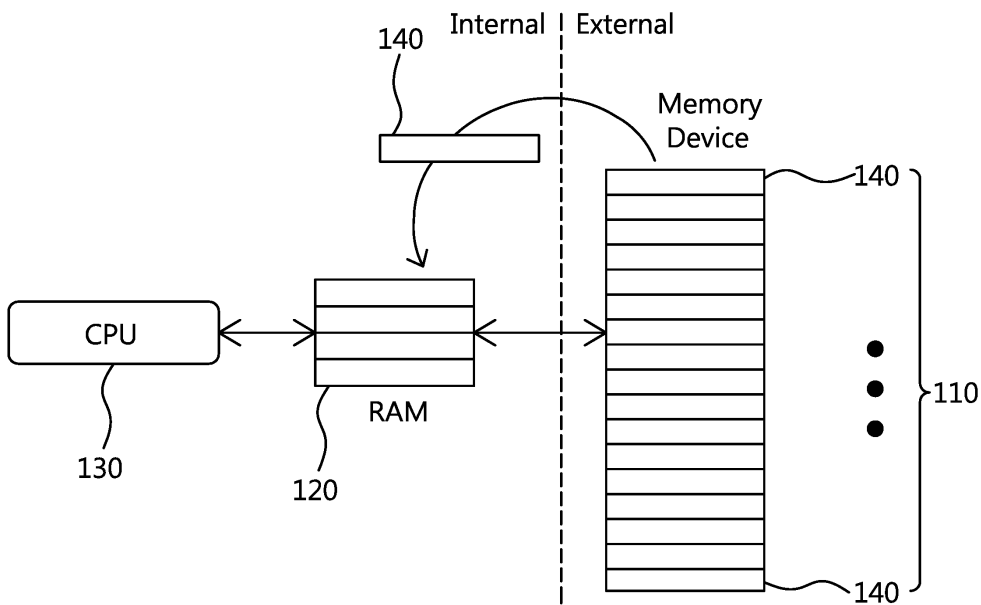
[0048] 이제까지 본 발명에 대하여 그 바람직한 실시예들을 중심으로 설명하였다. 그러나, 본 발명의 실시예는 당업계에서 통상의 지식을 가진 자에게 본 발명을 보다 완전하게 설명하기 위하여 제공되어지는 것으로, 본 발명의 범위가 상기의 실시예에 한정되는 것은 아니며, 여러 가지 다른 형태로 변형이 가능함은 물론이다.

부호의 설명

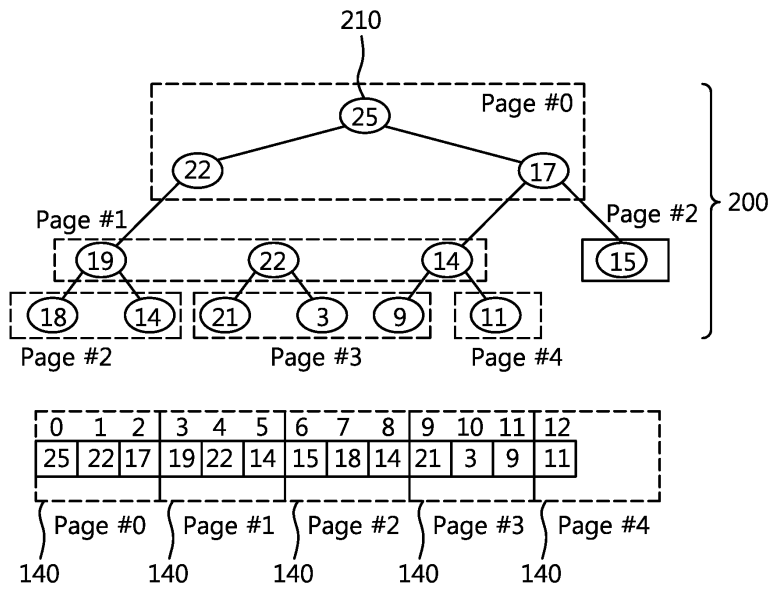
- | | | |
|--------|--------------------|----------------|
| [0049] | 110: 외부 메모리 장치 | 120: 메인메모리 |
| | 130: 중앙처리장치 | 140: 기본 액세스 단위 |
| | 200: 이진트리 | 210: 루트노드 |
| | 220: 서브트리 | 230: 레벨단위 |
| | 400: 힙 정렬 장치 | 410: 힙 트리 구성부 |
| | 420: 기본 액세스 단위 측정부 | 430: 서브트리 분할부 |
| | 440: 판별부 | 450: 레벨단위 분할부 |

도면

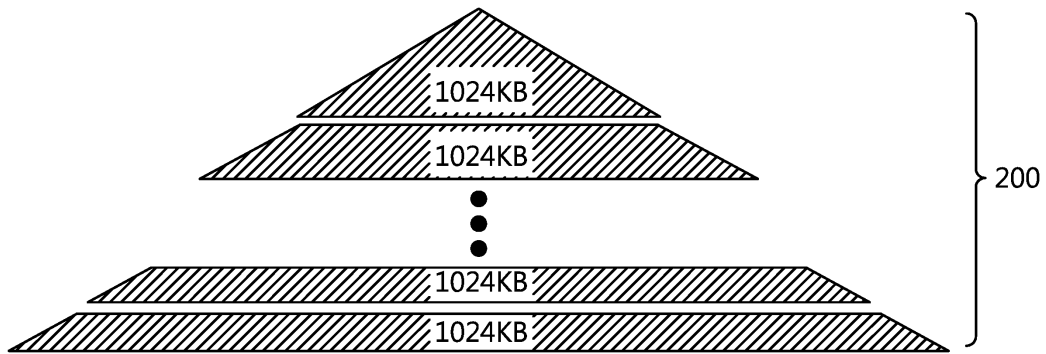
도면1



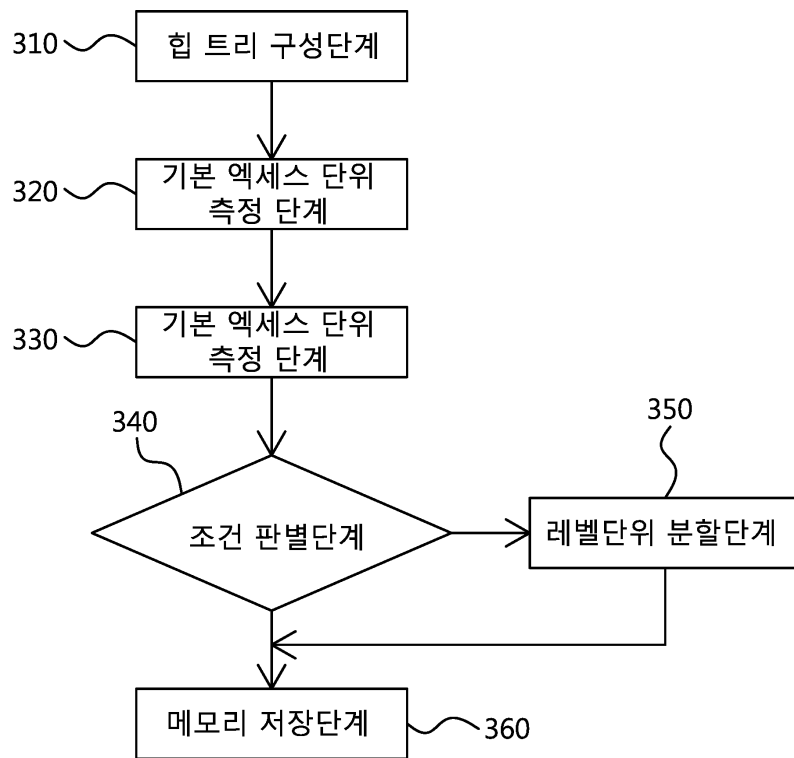
도면2a



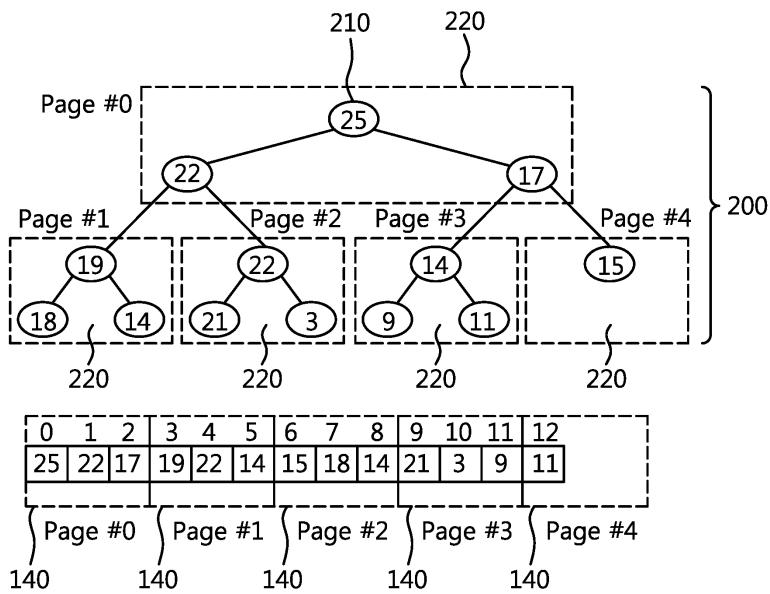
도면2b



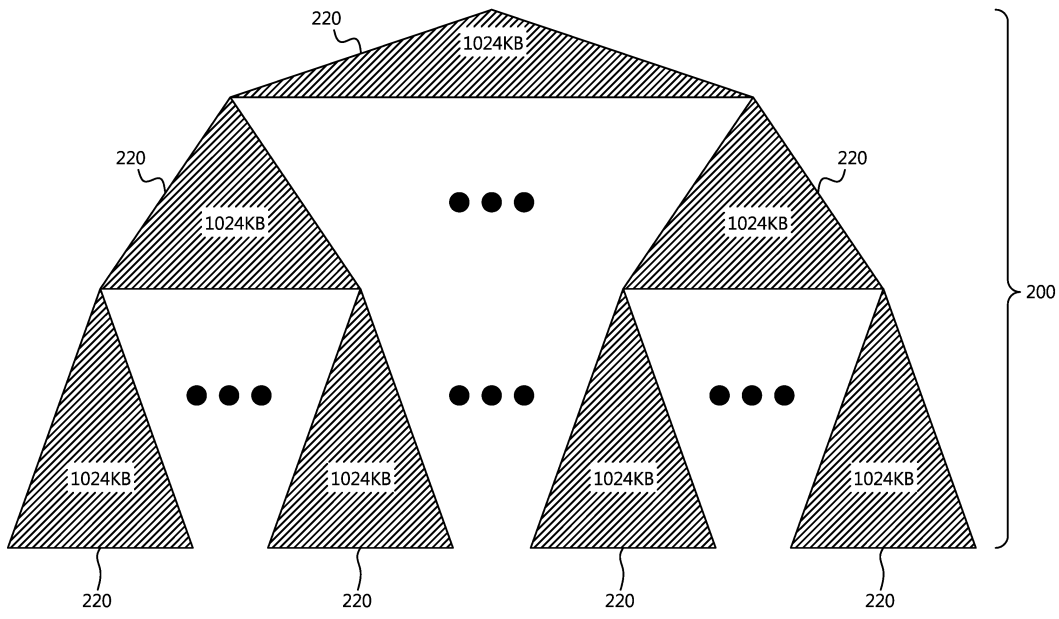
도면3



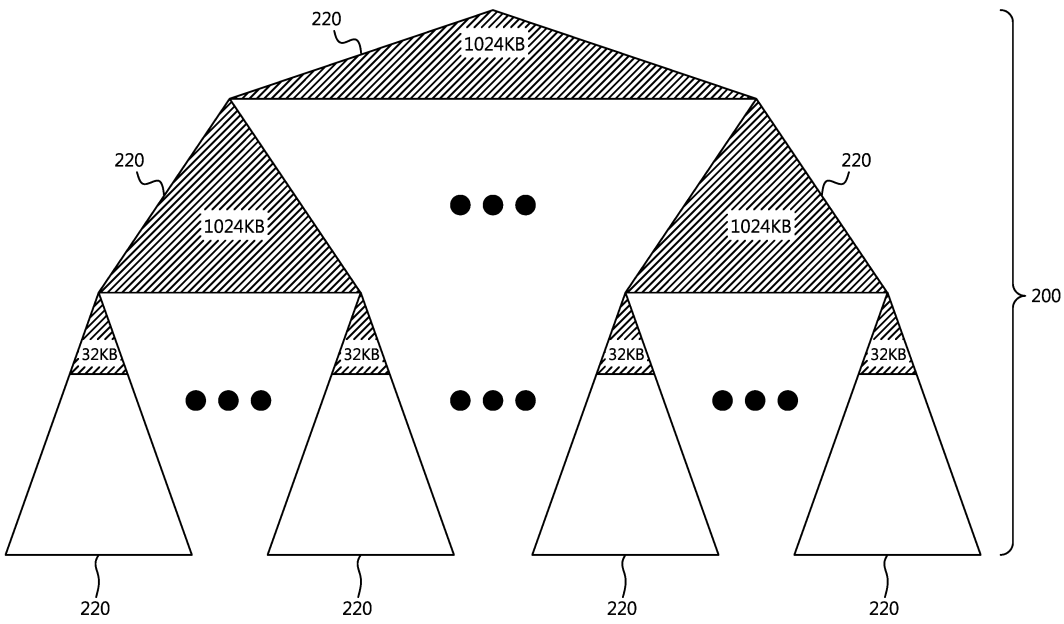
도면4a



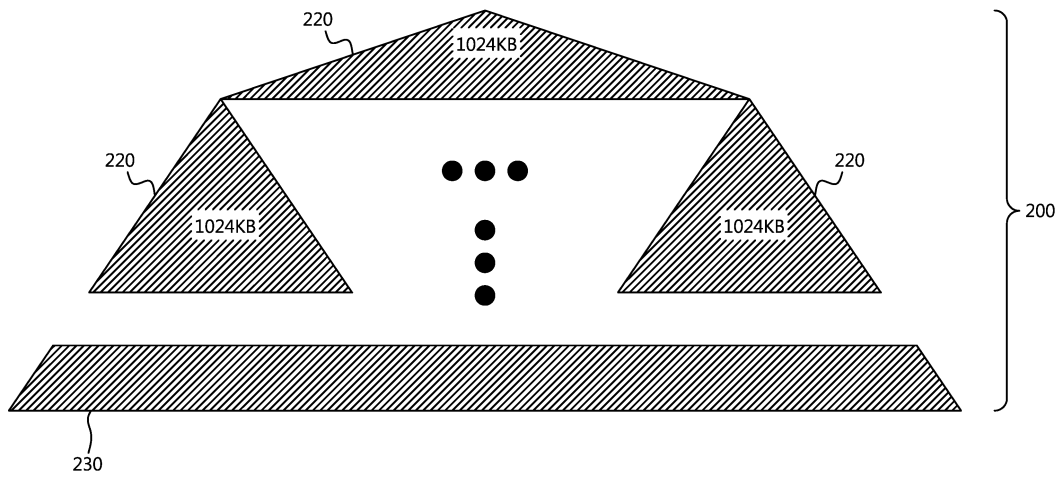
도면4b



도면5



도면6



도면7

