자료구조론 실습
Heap Sort

2017. 05. 24

한양대학교

이주홍

Heap이란?

- 자식보다 자기 자신의 **우선순위가 높은** 완전이진트리

- 우선순위: Max(값이 큰 게 우선)

  Min(값이 작은 게 우선)

  알파벳 순서 => priority(C) > priority(F)

  ...

Max-Heap Structure

```
#define MAX_ELEMENTS 100000
#define HEAP_FULL(n) (n == MAX_ELEMENTS- 1)
#define HEAP_EMPTY(n) (!n)
typedef struct {
    int key;
    /* other field */
} element; // heap의 각 노드 단위는 element라는 구조체
element heap[MAX_ELEMENTS]; // heap이라는 이름의 배열 10만개 생성
int n = 0; // heap의 사이즈
```

* ppt에 오타(MaX_ELEMENTS)

- 힙의 최대 크기(MAX_ELEMENTS)는 넉넉하게 100,000으로
- 원소(노드)는 element라는 구조체이지만, 어차피 int 하나만 사용하기 때문에 element -〉 int 로 바꿔서 과제를 진행해도 무방.

Max-Heap Insertion

```
void insert_max_heap(element item, int *n) {
    int i; // insert될 위치를 찾아줄 변수
    if (HEAP_FULL(*n)) { // Heap이 가득차면 넣을 수 없어
        fprintf(stderr,"The heap is full. \n");
        exit(1);
    }
    i = ++(*n); // 일단 insert하니까 사이즈(n)을 1증가시킴
    while ((i!=1 )&& (item.key > heap[i/2].key)) {
        heap[i] = heap[i/2];
        i /= 2;
    } // while: 부모를 계속 타고 올라가며, 새로 들어온 원소(item)의 위치(i)를 찾는 과정
    heap[i] = item; // 그렇게 해서 찾은 item의 위치(i)에 item을 저장
}
```

Max-Heap Deletion

```
element delete_max_heap(int *n) {
    element item, temp;
    if (HEAP_EMPTY(*n)) {
        fprintf(stderr,"The heap is empty\n");
        exit(1);
    }
    item = heap[1];
    temp = heap[(*n)--];
    parent = 1; child = 2;
    while (child <= *n) {
        /* compare left and right child's key values */
        if ((child < *n) && (heap[child].key <
                             heap[child+1].key))
            child++;
        if (temp.key >= heap[child].key) break;
        /* move to the next lower level */
        heap[parent] = heap[child];
        parent = child;
        child *= 2;
    }
    heap[parent] = temp;
    return item;
}
```

// heap의 루트를 제거하고 해당 값을 리턴

// heap이 비었는지 확인

// 우리 과제와 맞지 않음. 적절한 값을 리턴하도록 수정

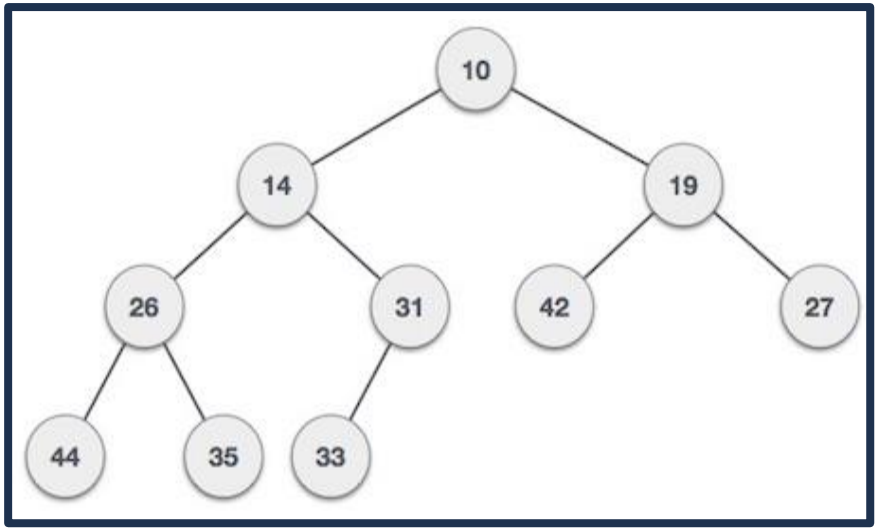
// 루트(heap[1])를 삭제하기 전에 미리 item에 빼둠

// 적절한 위치로 옮기기 위해 가장 마지막 값을 빼고 사이즈 1축소

// while: 끝에서 뺀 값을 루트에서부터 자식들을 확인하며 아래로 내리다가 힙 조건에 맞는 적절한 위치를 찾으면 stop

찾은 위치(parent)에 제일 끝에서 뺀 값(temp)를 저장하고, 초기 루트값인 item을 리턴

Heap Sort 원리

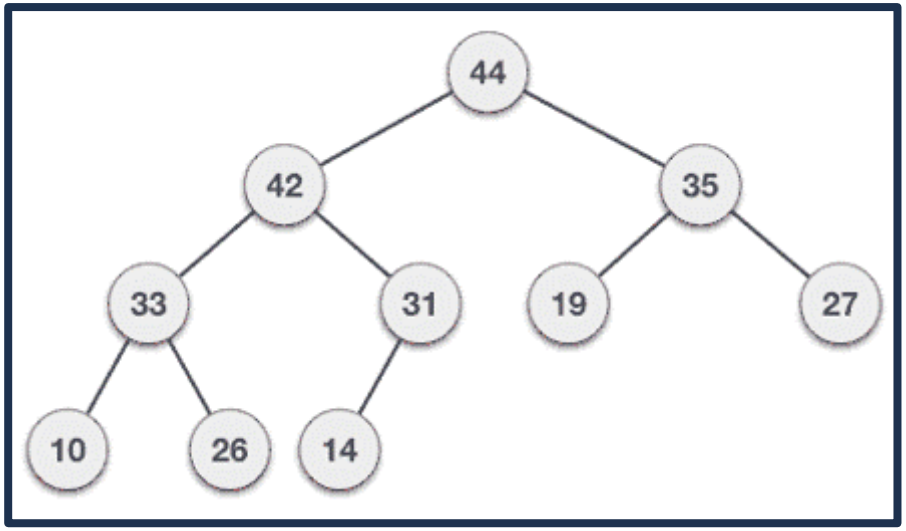


Min-Heap

Delete 10번

10	14	19	26	27	31	33	35	42	44

오름차순 정렬



Max-Heap

Delete 10번

44	42	35	33	31	27	26	19	14	10

내림차순 정렬

Heap Sort 구현

adjust() : root를 기준으로 하위 트리에 대해서 Heapify

```
void adjust(element list[], int root, int n) {
   int child, rootkey;
   element temp;
   temp = list[root];
   rootkey = list[root].key;
   child = 2 * root; /* left child */
   while(child <= n) {
      if ((child < n) && (list[child].key < list[child+1].key))
         child++;
      if (rootkey > list[child].key)
         break;
      else {
         list[child/2] = list[child];
         child *= 2;
      }
   }
   list[child/2] = temp;
}
void heapsort(element list[], int n) {
/* perform a heapsort on the array */
   int i, j;
   element temp;
   for (i = n / 2; i > 0; i--) /* initial heap construction */
       adjust(list, i, n);
   for (i = n - 1; i > 0; i--) { /* heap adjust */
       SWAP(list[1], list[i+1], temp);
       adjust(list, 1, i);
   }
}
```

기준

// root의 값이 자식들보다 크면 break!

// 자식이 더 크면 자식을 올리고
root는 내려가고

// 즉, 큰 값이 올라간다? => Max-Heap

// n/2부터 시작해서 0번째까지 전부 adjust

// 끝 원소와 root를 swap & root를 기준 adjust
// 마치 Deletion과 같다!

Assignment 7 - 순위표 만들기 2

- 학생 n명의 이름과 국어, 영어, 수학 점수가 주어질 때,
  학생들의 성적을 정렬하여 순위표를 만든다.

- 정렬 기준은 다음과 같다.
  1. 국어 점수가 감소하는 순서로
  2. 국어 점수가 같으면 영어 점수가 증가하는 순서로
  3. 국어 점수와 영어 점수가 같으면 수학 점수가 감소하는 순서로
  4. 모든 점수가 같으면 이름이 사전 순으로 증가하는 순서로 (대문자 -〉 소문자)
     - Ex) apple -〉 banana, banana -〉 bit, bit -〉 bitch, Sing -〉 apple

- 반드시 Heap Sort를 사용한다.

Assignment 7 – 순위표 만들기 2

- **입출력 설명**
  - 학생의 수 n($\leq 100{,}000$)이 첫째 줄에 입력된다.
  - 둘째 줄부터 n+1째 줄에 걸쳐 n명의 이름과 국영수 점수가 주어진다.
    각 점수는 1보다 크거나 같고, 100보다 작거나 같은 자연수이다.
  - 순위대로 학생의 이름을 한 줄에 하나씩 출력한다.

Assignment 7 - 순위표 만들기 2

입력 예시

학생의 수 n

n(=12)명의 점수

```
12
Junkyu 50 60 100
Sangkeun 80 60 50
Sunyoung 80 70 100
Soong 50 60 90
Haebin 50 60 100
Kangsoo 60 80 100
Donghyuk 80 60 100
Sei 70 70 70
Wonseob 70 70 90
Sanghyun 70 70 80
nsj 80 80 80
Taewhan 50 60 90
```

출력 예시

```
Donghyuk
Sangkeun
Sunyoung
nsj
Wonseob
Sanghyun
Sei
Kangsoo
Haebin
Junkyu
Soong
Taewhan
```

점수가 같으면 이름(사전 순)

Assignment 7 – 순위표 만들기 2

- **제출방식:** Assignment7 폴더 만들고
  Assignment7_학번.c 파일 저장
  Ex) Assignment7/Assignment7_2016000000.c

- **GitLab(https://hconnect.hanyang.ac.kr/)으로 제출**
- **GitLab이 안되면 이메일(roomylee@naver.com) 제출**

- **제출기한: 6월 1일 23시 59분**

감사합니다