

자료구조응용

14. Trees : heap

1. 다음 입력파일의 데이터를 사용하여 최대힙(Max Heap)에 대한 실습을 수행한다.

input.txt : 10 40 30 5 12 6 15 9 60

(1) 실행순서

① 파일입력을 받으면서 최대힙을 구성한다.

매 입력마다, 구성된 최대힙의 배열원소를 인덱스 순서대로 출력한다.

② 최대힙의 최대값을 연속으로 원소개수만큼 삭제한다.

매 삭제마다, 재구성된 최대힙의 배열원소를 인덱스 순서대로 출력한다.

(3) 실행 예

```
C:\WINDOWS\system32\cmd...
***** insertion into a max heap *****
10
40 10
40 10 30
40 10 30 5
40 12 30 5 10
40 12 30 5 10 6
40 12 30 5 10 6 15
40 12 30 9 10 6 15 5
60 40 30 12 10 6 15 5 9
***** deletion from a max heap *****
delet data = 60
40 12 30 9 10 6 15 5
delet data = 40
30 12 15 9 10 6 5
delet data = 30
15 12 6 9 10 5
delet data = 15
12 10 6 9 5
delet data = 12
10 9 6 5
delet data = 10
9 5 6
delet data = 9
6 5
delet data = 6
5
delet data = 5
계속하려면 아무 키나 누르십시오 . . .
```

2. 다음 입력파일의 데이터를 사용하여 최소힙(Min Heap)에 대한 실습을 수행한다.

input.txt : 10 40 30 5 12 6 15 9 60

(1) 실행순서

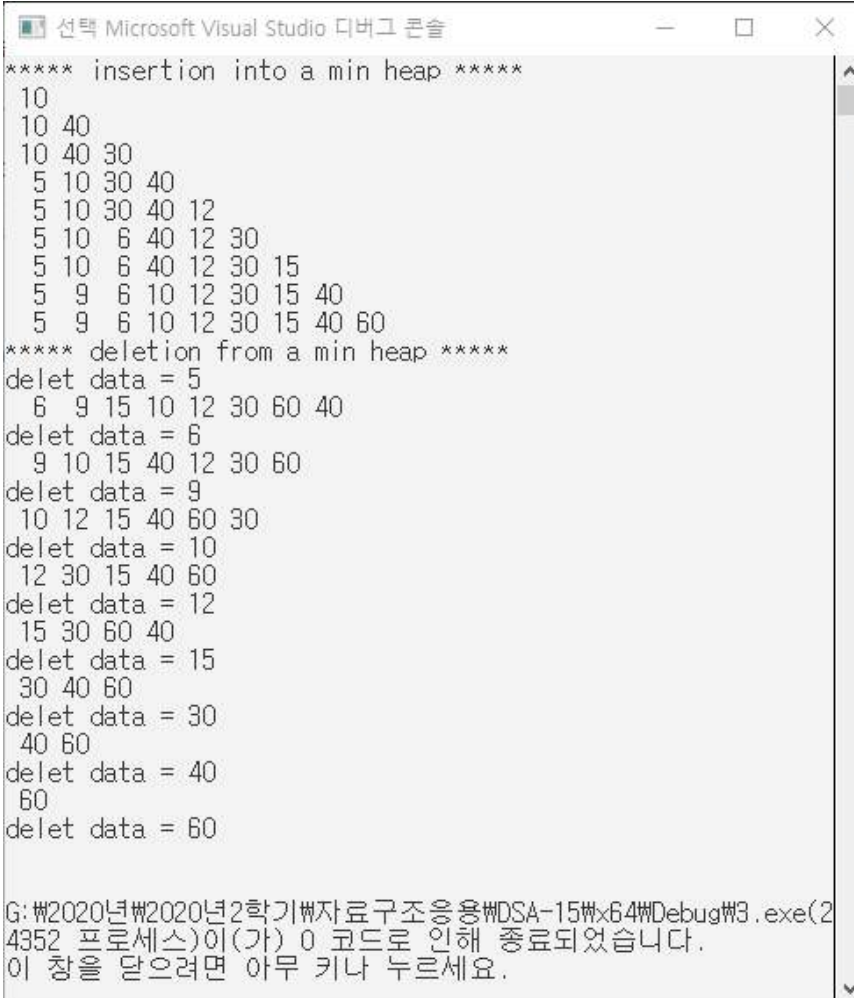
① 파일입력을 받으면서 최소힙을 구성한다.

매 입력마다, 구성된 최소힙의 배열원소를 인덱스 순서대로 출력한다.

② 최소힙의 최소값을 연속으로 원소개수만큼 삭제한다.

매 삭제마다, 재구성된 최대힙의 배열원소를 인덱스 순서대로 출력한다.

(3) 실행 예



```
***** insertion into a min heap *****
10
10 40
10 40 30
5 10 30 40
5 10 30 40 12
5 10 6 40 12 30
5 10 6 40 12 30 15
5 9 6 10 12 30 15 40
5 9 6 10 12 30 15 40 60
***** deletion from a min heap *****
delet data = 5
6 9 15 10 12 30 60 40
delet data = 6
9 10 15 40 12 30 60
delet data = 9
10 12 15 40 60 30
delet data = 10
12 30 15 40 60
delet data = 12
15 30 60 40
delet data = 15
30 40 60
delet data = 30
40 60
delet data = 40
60
delet data = 60

G:\2020년\2020년2학기\자료구조\응용\DSA-15\64\Debug\3.exe(2
4352 프로세스)이(가) 0 코드로 인해 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.
```

3. 다음과 같이 임의의 노드 n개로 구성된 이진탐색트리(binary search tree)를 생성하여 아래와 같이 실행하도록 프로그램을 작성하라.

(1) 실행순서

① 난수생성을 위한 seed와 이진탐색트리의 노드 개수(n)를 입력받음

```
printf("random number generation (1 ~ %d)\n", MAX_SIZE);
printf("%s", "the number of nodes in BST (less than and equal to 50) : " );
scanf_s("%d", &n);
printf("%s", "seed : " );
scanf_s("%u", &seed);
printf("\ncreating a BST from random numbers\n");
srand(seed);
```

② 1~500 범위의 난수를 생성하여 노드의 key와 item 필드 값으로 동일하게 사용(키값과 항목 값이 같음), 난수가 발생 되는 순서대로 출력 할 것

※ 이진탐색트리의 key 값은 중복이 허용되지 않음을 주의

③ ②의 key, item을 사용하여 이진탐색트리에 노드를 하나 추가함

④ ②~③ 과정을 n번 수행하여 이진탐색트리를 구성

※ 난수발생 순서대로 노드를 추가해야 함

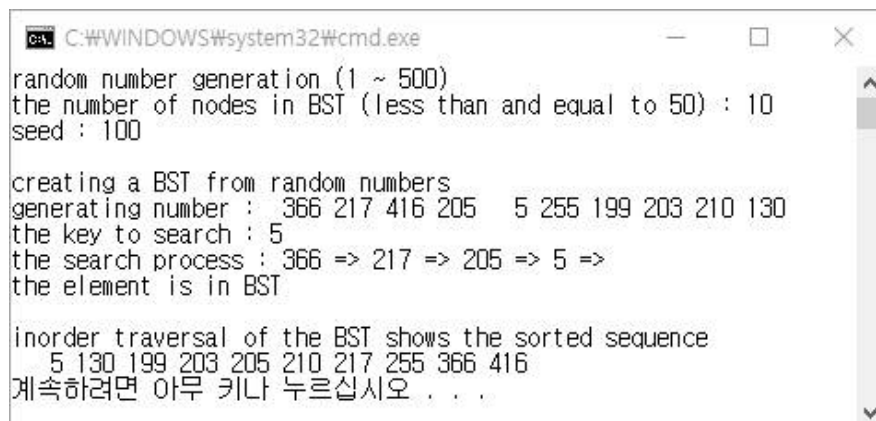
⑤ 탐색할 key를 입력받아서 이진탐색하여 그 결과를 출력한다.

탐색과정을 출력하시오

⑥ 이진탐색트리를 구성하고 있는 노드의 key값을 오름차순으로 정렬되도록 출력함

※ inorder traversal 사용

(2) 실행 예



```
C:\WINDOWS\system32\cmd.exe
random number generation (1 ~ 500)
the number of nodes in BST (less than and equal to 50) : 10
seed : 100

creating a BST from random numbers
generating number : 366 217 416 205 5 255 199 203 210 130
the key to search : 5
the search process : 366 => 217 => 205 => 5 =>
the element is in BST

inorder traversal of the BST shows the sorted sequence
5 130 199 203 205 210 217 255 366 416
계속하려면 아무 키나 누르십시오 . . .
```

■ 제출 형식

- 솔루션 이름 : DS_14
- 프로젝트 이름 : 1, 2, 3
- 솔루션 폴더를 압축하여 제출할 것.
- 학습관리시스템에 과제를 올릴 때 제목:
1차 제출: 학번_이름_DS_14(1), 2차 제출: 학번_이름_DS_14(2)
제출은 2회 걸쳐 가능(수정 시간 기준으로 처리)