**과제1**

**16010946 김은숙**

**소스코드 :**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

//자리바꾸는 함수

void swap(int \*x, int \*y) {

int temp;

temp = \*x;

\*x = \*y;

\*y = temp;

}

//다운힙 함수

void downHeap(int L[],int i,int n) {

int left = i \* 2;

int right = i \* 2 + 1;

int bigger = left;

//자식 노드가 없는지 확인

if (left> n) {

return;

}

//크기 비교하여 자리바꿈

if (right <= n) {

if (L[right] > L[bigger]) {

bigger = right;

}

}

if (L[i] >= L[bigger])return;

swap(&L[i], &L[bigger]);

//재귀하여 내려가면서 자리를 찾아간다.

downHeap(L,bigger,n);

}

//힙 생성 함수

void rBuildHeap(int L[],int i,int n) {

if (i > n) {

return;

}

//왼쪽 자식과 오른쪽 자식으로 재귀하여 내려간 후 downheap 함수로 자리를 힙에 맞는 자리를 찾아간다.

rBuildHeap(L,2 \* i,n);

rBuildHeap(L,2 \* i + 1,n);

downHeap(L,i,n);

return;

}

//해당 순위의 원소 찾는 함수

int findKthSmallest(int L[], int n, int k) {

int i;

//전부 정렬할 필요 없으므로 k번째까지 정렬

for (i = n; i >= k; i--) {

//이미 정렬이 되어있는 경우를 조건문으로 거른다.

if (L[i] < L[1]) {

swap(&L[1], &L[i]);

downHeap(L, 1, i - 1);

}

}

return L[k];

}

//main

void main() {

//n : 리스트크기,i : 반복문 및 인덱스 ,k : 순위, \*L : 배열

int n, i, k, \*L;

//리스트 크기 입력

printf("리스트 크기:");

scanf("%d", &n);

//동적할당

L = (int\*)malloc(sizeof(int)\*(n+1));

//난수 발생

srand(time(NULL));

for (i = 1; i <= n; i++) {

L[i] = rand() % 1000000 + 1;

}

//충분히 작은 수일 경우 리스트 출력

if (n <= 20) {

printf("리스트:");

for (i = 1; i <= n; i++) {

printf(" %d", L[i]);

}

printf("\n");

}

//힙생성

rBuildHeap(L,1,n);

for (i = 0; i < 3; i++) {

//순위 입력

printf("순위:");

scanf("%d", &k);

//해당 순위의 원소 출력

printf("원소:%d\n", findKthSmallest(L, n, k));

}

//동적할당 해제

free(L);

}

**캡쳐 이미지 :**

