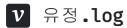
23. 9. 19. 오후 10:30 Week3 과제





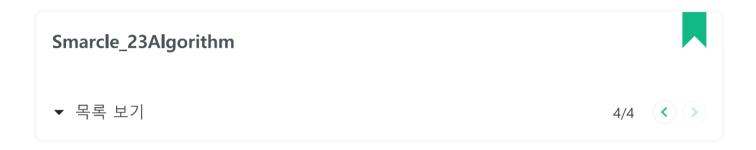
Week3_과제

통계 수정 삭제

yuyu7123 · 방금 전 · 비공개



2075 힙정렬



Baekjoon_2075 : N번째 큰 수

힙과 힙정렬을 정의하고 NxN개의 값을 입력 받아 정렬한 뒤 N번째 값 출력

풀이 과정

- 1. 힙정렬에 쓰이는 upHeap, downHeap, buildHeap 함수를 정의하여 제자리 힙정렬 사용
- 2. NxN개의 값을 입력한 뒤 힙에 저장
- 3. 힙정렬 함수 호출
- 4. N번째 힙에 저장된 값 출력

문제 해결

• 오름차순 정렬 뒤에서 원하는 순서의 값을 빼는 것으로 해결

코드

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>

typedef struct Node {
```

```
23. 9. 19. 오후 10:30
        int key;
    } Node;
    typedef struct Heap {
        Node* node;
        int size;
        int capacity;
    } Heap;
    Heap* heap;
    Heap* init_Heap(int capacity) {
        Heap* heap = (Heap*)malloc(sizeof(Heap));
        heap->capacity = capacity;
        heap->size = 0;
        heap->node = (Node*)malloc(sizeof(Node) * (capacity + 1));
        return heap;
    }
    void upHeap(int index) {
        Node temp;
        if (index == 1) return;
        if (heap->node[index].key >= heap->node[index / 2].key) {
            temp = heap->node[index];
            heap->node[index] = heap->node[index / 2];
            heap->node[index / 2] = temp;
            index = index / 2;
            upHeap(index);
        }
    }
    void downHeap(int i, int last) {
        int left = 2 * i, right = 2 * i + 1;
        if (left > last) return;
        int larger = left;
        if (right <= last) {</pre>
            if (heap->node[right].key > heap->node[larger].key)
                 larger = right;
        }
        if (heap->node[i].key >= heap->node[larger].key) return;
        Node temp = heap->node[i];
        heap->node[i] = heap->node[larger];
        heap->node[larger] = temp;
        downHeap(larger, last);
    }
    void insertItem(int key) {
        heap->size++;
        int index = heap->size;
        heap->node[index].key = key;
        upHeap(index);
    }
```

```
void rBuildHeap(int i) {
    if (i > heap->capacity) return;
    rBuildHeap(2 * i);
    rBuildHeap(2 * i + 1);
    downHeap(i, heap->capacity);
}
void buildHeap() {
    for (int i = heap->capacity / 2; i > 0; i--) downHeap(i, heap->capacity);
}
void inPlaceHeapSort() {
    //rBuildHeap(1);
    buildHeap();
    for (int i = heap \rightarrow capacity; i > 1; i--) {
        Node temp = heap->node[1];
        heap->node[1] = heap->node[i];
        heap->node[i] = temp;
        downHeap(1, i - 1);
    }
}
int main() {
    int n, m;
    scanf("%d", &n);
    heap = init_Heap(n*n);
    for (int i = 0; i < n*n; i++) {
        scanf("%d", &m);
        insertItem(m);
    }
    inPlaceHeapSort();
    printf("%d ", heap->node[n*n-n+1].key);
    free(heap->node);
    free(heap);
    return 0;
}
```



유정

23. 9. 19. 오후 10:30 Week3_과제



0개의 댓글

댓글을 작성하세요

댓글 작성

