

2023_Autumn_Algorithm_Study_3주차 과제 기록

통계 수정 삭제

jaesung2da · 방금 전 · 비공개

❤ 0

3주차 과제는
2075번: N번째 큰 수

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
1 초	12 MB (하단 참고)	26532	10631	7700	39.005%

문제

$N \times N$ 의 표에 수 N^2 개 채워져 있다. 채워진 수에는 한 가지 특징이 있는데, 모든 수는 자신의 한 칸 위에 있는 수보다 크다는 것이다. $N=5$ 일 때의 예를 보자.

12	7	9	15	5
13	8	11	19	6
21	10	26	31	16
48	14	28	35	25
52	20	32	41	49

이러한 표가 주어졌을 때, N 번째 큰 수를 찾는 프로그램을 작성하시오. 표에 채워진 수는 모두 다르다.

입력

첫째 줄에 $N(1 \leq N \leq 1,500)$ 이 주어진다. 다음 N 개의 줄에는 각 줄마다 N 개의 수가 주어진다. 표에 적힌 수는 -10억보다 크거나 같고, 10억보다 작거나 같은 정수이다.

출력

첫째 줄에 N 번째 큰 수를 출력한다.

예제 입력 1 복사

```
5
12 7 9 15 5
13 8 11 19 6
21 10 26 31 16
48 14 28 35 25
52 20 32 41 49
```

예제 출력 1 복사

```
35
```

저번 2주차 과제와 스터디를 하면서 느낀점은 문제 해결 시간과 메모리 사용량을 줄이면서 수정에 용이한 코드가 좋은 코드라는 생각이 들었다.

문제 풀기 전 전략

위에서 입력으로 받은 수를 모두 배열이나 다른 형태로 저장하는 것은 메모리를 많이 잡아 먹을 것이라고 판단했다. 그리고 결국 원하는 것은 N 번째 큰 수만 구하면 되기 때문에 전체를 저장하지 않고 입력을 받았을 때 바로 N 번째를 출력하는 것이 중요하다고 생각했다. 설계한 전략은 우선순위 큐, 힙을 이용하는 것이 좋다고 생각했다. 그 중에서도 **최소힙**을 사용하면 N 개의 힙만을 이용해서 문제를 풀 수 있다고 생각했다. N 번째 큰 수를 고르는 것이기 때문에 최대힙을

사용해야 하는 것이 아닌가 잠깐 생각했지만 최대힙을 사용하면 정렬은 루트에 가장 큰 수가 오지만 데이터를 삭제할 때 가장 큰 값 -> 루트를 먼저 없애기 때문에 최소힙을 사용해서 값을 점점 대입과 삭제를 반복하여 작은 수들을 없애고자 한다.

그렇게 된다면 결과적으로 마지막 루트에 있는 값이 N번 째 큰수가 된다!

최소힙을 구현하기 위해서 알고리즘 실습시간에 제출했던 oj에 최대힙 코드를 기반으로 수정해나갔다.

```
#pragma warning(disable:4996)
#include <stdio.h>
#include <stdlib.h>

int Heap[1500];
int heapSize = 0;

void upHeap(int idx) {
    int tmp;
    int Parent = idx / 2;
    if (idx == 1 || Heap[Parent] <= Heap[idx]) return;
    tmp = Heap[Parent];
    Heap[Parent] = Heap[idx];
    Heap[idx] = tmp;
    upHeap(Parent);
}

void downHeap(int idx) {
    int Leftchild_idx = 2 * idx, Rightchild_idx = 2 * idx + 1;
    int smaller_idx, tmp;
    if (Leftchild_idx > heapSize) return;
    smaller_idx = Leftchild_idx;
    if (Rightchild_idx <= heapSize) {
        if (Heap[smaller_idx] > Heap[Rightchild_idx]) smaller_idx = Rightchild_idx;
    }
    if (Heap[smaller_idx] < Heap[idx]) {
        tmp = Heap[smaller_idx];
        Heap[smaller_idx] = Heap[idx];
        Heap[idx] = tmp;
        downHeap(smaller_idx);
    }
}

void insert(int key) {
    heapSize++;
    Heap[heapSize] = key;
    upHeap(heapSize);
}

int removeMin() {
    int root = Heap[1];
    Heap[1] = Heap[heapSize--];
    downHeap(1);
    return root;
}

int main() {
    int N;
    scanf("%d", &N);
```

```

int key;
for (int i = 0; i < N; i++) {
    for (int j = 0; j < N; j++) {
        scanf("%d", &key);
        insert(key);
        if (heapSize > N)
            removeMin();
    }
}

printf("%d\n", removeMin());

return 0;
}

```

제출 번호	아이디	문제	결과	메모리	시간	언어	코드 길이	제출한 시간
67154172	tlawotjd	2075	맞았습니다!!	1120 KB	572 ms	C99 / 수정	1165 B	1분 전

소요시간 : 약 2시간...

양심선언 : 약속시간 임박으로 코드는 다 짰지만 자꾸 엉뚱한 출력이 나와서 GPT한테 뭐가 문제인지 물어봄,,,,

원래는 insert함수 와 remove함수를 합쳐서 작성했는데 분리해주니 성공했다...



심재성



이전 포스트

2023_Autumn_Algorithm_Study_2주차 과제 기록

0개의 댓글

댓글을 작성하세요

