

- 1-1.c

- 알고리즘
 1. Input 을 받는 순서대로 maxHeap 에 insert 한다.
 2. HeapSort 를 통해 재배치한다.
- 코드설명

Main 함수

```
30 int main(void) {
31     PriorityQueue Q;
32     Init(&Q, 99);
33
34     int num;
35     int i;
36     for(i=1; i<100; i++){
37         scanf("%d", &num);
38         if(num == -1)
39             break;
40         //정렬할 값을 순서 상관없이 입력받는대로 배열을 사용하여 완전 이진 트리에 배치한다
41         Q.Elements[i] = num;
42     }
43
44     Q.Size = i-1; //사이즈를 설정해준다
45     HeapSort(Q.Elements, Q.Size); //마지막노드부터 이진 힙의 규칙에 맞도록 재배치한다
46
47     int j;
48     for(j=0; j<i-1; j++){
49         //마지막노드와 자리를 바꿔 print를 한뒤
50         //사이즈를 줄이고 다시 재배치를 하며 끝까지 반복한다
51         printf("%d ", DeleteMin(&Q));
52     }
53     printf("\n");
54
55     return 0;
56 }
```

주요 함수

```
82 int DeleteMin(PriorityQueue *H){
83     int i, Child;
84     int MinElement, LastElement;
85
86     MinElement = H->Elements[1];
87     LastElement = H->Elements[H->Size--];
88
89     //PercDown
90     for(i=1; i*2<=H->Size; i=Child){
91         Child = i*2;
92         if(Child != H->Size && H->Elements[Child + 1] > H->Elements[Child])
93             Child++;
94         if(LastElement < H->Elements[Child])
95             H->Elements[i] = H->Elements[Child];
96         else
97             break;
98     }
99     H->Elements[i] = LastElement;
100    return MinElement;
101 }
102
103 void HeapSort(int A[], int N){
104     int i;
105     for(i = N/2; i>0; i--)
106         PercDown(A, i, N);
107     for(i = N; i>0; i--){
108         Swap(&A[1], &A[i]);
109         PercDown(A, 1, i-1);
110     }
111 }
112 //A[left] ~ A[right]를 max힙으로 만드는 함수
113 void PercDown(int A[], int left, int right){
114     int temp = A[left]; //루트 노드
115     int Parent, Child;
116
117     for(Parent = left; Parent*2 <= right; Parent = Child){
118         int cl = Parent*2; //왼쪽 자식
119         int cr = cl+1; //오른쪽 자식
120         Child = (cr <= right && A[cr] < A[cl]) ? cr : cl; //작은값을 선택
121         if(temp < A[Child])
122             break; //A[Parent] = A[Child];
123         else
124             A[Parent] = A[Child]; //break;
125     }
126     A[Parent] = temp;
127 }
```

Max Heap 과정

```
~/Downloads/2-2/알고/hw/project1/HW1_2016025187 — -bash
[gimdo-eun-ui-MacBook-Pro:HW1_2016025187 doeun$ gcc -o 1-1.exe 1-1.c
[gimdo-eun-ui-MacBook-Pro:HW1_2016025187 doeun$ ./1-1.exe
9 45 871 23 13 13 88 46 12 51 99 -1
871 99 88 51 46 45 23 13 13 12 9
j : 0, del : 871
99 51 88 13 46 45 23 9 13 12
j : 1, del : 99
88 51 45 13 46 12 23 9 13
j : 2, del : 88
51 46 45 13 13 12 23 9
j : 3, del : 51
46 13 45 9 13 12 23
j : 4, del : 46
45 13 23 9 13 12
j : 5, del : 45
23 13 12 9 13
j : 6, del : 23
13 13 12 9
j : 7, del : 13
13 9 12
j : 8, del : 13
12 9
j : 9, del : 12
9
j : 10, del : 9

gimdo-eun-ui-MacBook-Pro:HW1_2016025187 doeun$
```

- 컴파일환경 및 실행 화면

```
[gimdo-eun-ui-MacBook-Pro:HW1_2016025187 doeun$ gcc -o 1-1.exe 1-1.c
[gimdo-eun-ui-MacBook-Pro:HW1_2016025187 doeun$ ./1-1.exe
9 45 871 23 13 13 88 46 12 51 99 -1
871 99 88 51 46 45 23 13 13 12 9
gimdo-eun-ui-MacBook-Pro:HW1_2016025187 doeun$
```

• 1-2.c

- 알고리즘

1. 이중배열을 통해 input1-2.txt 값을 한번에 받아오면서,
각 배열의 원소의 개수의 총합을 구한다
2. 총 원소의 개수를 담을수 있는 minHeap 을 생성한다.
3. 각 row 에 담긴 배열을 column 기준으로 비교없이 insert 한다.
4. 만들어진 minHeap 을 정렬한다.

- 코드설명

함수 정의 및 struct

```
7 typedef tuple<int, int, int> Tuple; //value, k번째 배열, t번째 원소
8
9 //typedef struct HeapStruct PriorityQueue;
10 typedef struct HeapStruct{
11     int Capacity;
12     int Size;
13     vector<Tuple> Elements;
14 } PriorityQueue;
15
16 int IsFull(PriorityQueue *H) {
17     if(H->Size == (H->Capacity))
18         return 1;
19     else return 0;
20 }
21 void Swap(Tuple *a, Tuple *b){
22     Tuple tmp = *a;
23     *a = *b;
24     *b = tmp;
25 }
26
27 void Init(PriorityQueue *H, int capacity);
28 void Insert(PriorityQueue *H, Tuple X);
29 void HeapSort(vector<Tuple> A, int N);
30 void PercDown(vector<Tuple> A, int left, int right); //A[left]~A[right]를 min힙으로 만듦
31 Tuple DelTop(PriorityQueue *H);
32 void Print(PriorityQueue *H);
```

Main 함수

```
34 int main(void) {
35
36     FILE *input = fopen("input1-2.txt", "r"); //실행파일과 같은 경로상에 있는 input파일
37
38     int inputArr[99][99];
39     int i,j;
40     char c;
41     Tuple top;
42
43     for(i=0; i<99; i++){
44         for(j=0; j<99; j++){
45             inputArr[i][j] = inf; //초기화
46         }
47     }
48
49     int k;
50     fscanf(input, "%d", &k);
51     for(i=0; i<k; i++){
52         for(j=0; j<99; j++){
53             if(EOF!=fscanf(input, "%d%c", &inputArr[i][j], &c)){
54                 if(c == ' '){
55                     continue;
56                 }
57                 else if(c == '\n'){
58                     break;
59                 }
60             }
61         }
62     }
63
64     PriorityQueue Q;
65     Init(&Q, k);
```


Input 파일 k 개의 배열을 받아 k 개의 capacity 로 HeapSort 하는 과정

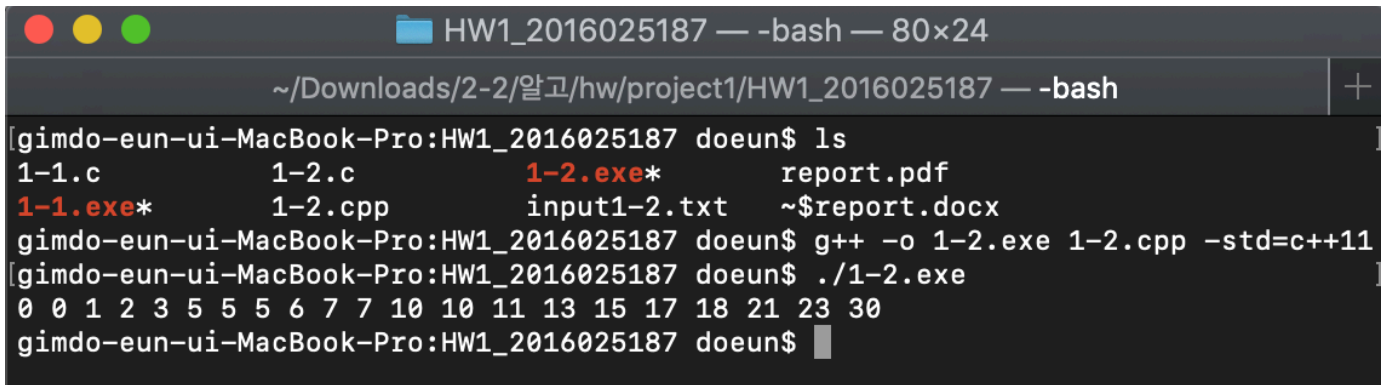
```

first Heap : 0 0 1 17
top : 0      Heap : 0 17 1
| ins : 2 >> Heap : 0 2 1 17
top : 0      Heap : 1 2 17
| ins : 5 >> Heap : 1 2 17 5
top : 1      Heap : 2 5 17
| ins : 7 >> Heap : 2 5 17 7
top : 2      Heap : 5 7 17
| ins : 3 >> Heap : 3 5 17 7
top : 3      Heap : 5 7 17
| ins : 5 >> Heap : 5 5 17 7
top : 5      Heap : 5 7 17
| ins : 6 >> Heap : 5 6 17 7
top : 5      Heap : 6 7 17
| ins : 5 >> Heap : 5 6 17 7
top : 5      Heap : 6 7 17
| ins :10 >> Heap : 6 7 17 10
top : 6      Heap : 7 10 17
| ins : 7 >> Heap : 7 7 17 10
top : 7      Heap : 7 10 17
| ins :11 >> Heap : 7 10 17 11
top : 7      Heap : 10 11 17
| ins :10 >> Heap : 10 10 17 11
top : 10     Heap : 10 11 17
| ins :30 >> Heap : 10 11 17 30
top : 10     Heap : 11 30 17
| ins :13 >> Heap : 11 13 17 30
top : 11     Heap : 13 30 17
top : 13     Heap : 17 30
| ins :15 >> Heap : 15 30 17
top : 15     Heap : 17 30
top : 17     Heap : 30
| ins :18 >> Heap : 18 30
top : 18     Heap : 30
| ins :21 >> Heap : 21 30
top : 21     Heap : 30
| ins :23 >> Heap : 23 30
top : 23     Heap : 30
top : 30     Heap :

```

Program ended with exit code: 0

- 컴파일환경 및 실행 화면



```
HW1_2016025187 — -bash — 80x24
~/Downloads/2-2/알고/hw/project1/HW1_2016025187 — -bash
[gimdo-eun-ui-MacBook-Pro:HW1_2016025187 doeun$ ls
1-1.c          1-2.c          1-2.exe*       report.pdf
1-1.exe*       1-2.cpp        input1-2.txt   ~$report.docx
[gimdo-eun-ui-MacBook-Pro:HW1_2016025187 doeun$ g++ -o 1-2.exe 1-2.cpp -std=c++11
[gimdo-eun-ui-MacBook-Pro:HW1_2016025187 doeun$ ./1-2.exe
0 0 1 2 3 5 5 5 6 7 7 10 10 11 13 15 17 18 21 23 30
[gimdo-eun-ui-MacBook-Pro:HW1_2016025187 doeun$
```