

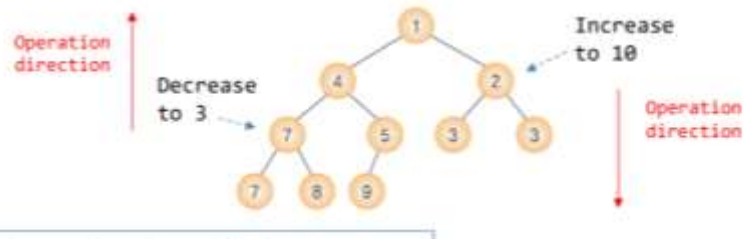
1 번

1. (Programming: 30 points)

Implement the 'decrease-key' and 'increase_key' in the min-heap, as explained in p53 of 'DS-Lec10-Graph'.

(Hint: refer to the insert operation in the min heap)

Test your code using the following input.



1) 문제해결

Decrease_key_min_heap 은 parent node 와 비교해가며 작은 값의 node 를 위로 올리는 함수이다.

i 번째 node 의 parent node 를 찾아서 비교 후 parent 가 더 큰 값이면 node 를 교환한다.

Increase_min_heap 은 child node 와 비교해가며 큰 값의 node 를 아래로 내리는 함수이다.

i 번째 node 의 child node 를 찾아서 비교 후 child 가 더 작은 값이면 node 를 교환한다.

2) 실행결과

```
1
3
3
4
5
10
3
7
8
9
```

2 번

문제

2. (Programming: 70 points)

In p55-56 of 'DS-Lec10-Graph', Prim algorithm was implemented using an unsorted array 'dist'. Revise this code by referring to p51.

- Use the min heap, not the unsorted array.

- Specify the parent and child relation inside the code.

- Use the following two functions.

'build_min_heap': use this at 'Insert all vertices into the priority queue Q'

'delete_min_heap': use this at 'Extract_Min(Q)'

'Decrease_key_min_heap': use this at ' $\text{dist}[v] \leftarrow \text{weight}[u][v]$ '

1) 문제해결

정렬되지 않은 행렬이 아닌 min heap 을 이용해서 Prim algorithm 을 구현하는 문제이다. Prim algorithm 은 첫 vertex 에서 시작해서 mst 를 만들어 나간다. 처음에 모든 vertex 에 무한대 숫자를 넣고 edge 들과 비교해가며 더 작은 값을 vertex 에 넣는다. Min heap 으로 구현하기 위해서 Kruskal Algorithm 에서 사용한 방법을 일부분 가져온다. Heap 에 연결된 node 들도 저장하고 min heap 으로 정렬한다 이진트리로 만들어진 heap 을 preorder 순으로 꺼내서 MST 를 만든다.

2) 실행결과

```
vertex 0 -> 1 edge: 3
vertex 1 -> 2 edge: 8
vertex 2 -> 3 edge: 15
vertex 2 -> 4 edge: 2
vertex 4 -> 5 edge: 9
vertex 4 -> 6 edge: 4
vertex 4 -> 7 edge: 5
```