



H.w2 Min heap.

Implement Dijkstra's algorithm with the help of min heap.

Routing Table

Graph

bool visited
int cost
int from

input:

6 ← source node

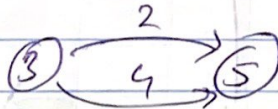
11 ← #nodes 0...10

31 ← # edges (directed)

There is an edge from
2 to Node 7 with a
cost of 8

2 7 8
 0 3 11
 2 6 5
 } 31 lines

- It is possible that there are multiple edges from a node to another.



- Three structures are needed.

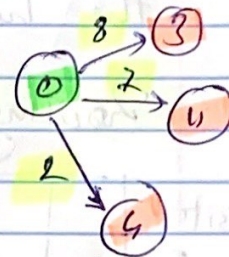
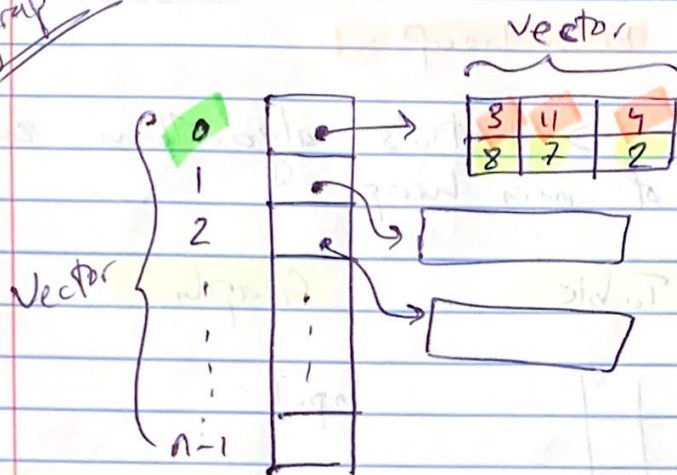
① Routing table

Vector <rt.node>

② graph

bool visited —
 int cost —
 int from —
 int heap position —

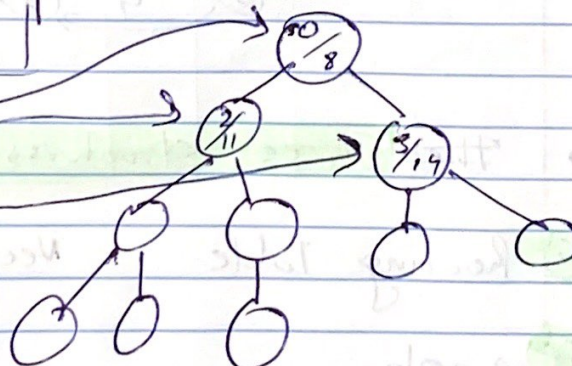
graph



3. Heap.

Routing Table.

	0	1	2	3	4	...
Node visited						
int cost	8	21	11	14	19	
int from						
int heap						
-position						



0	1	2	3	4	5
0	2	3	-	-	-

Output - O/P

The cost From node 7 to node 0 is 35; from is node 8.

```
class heap-node {  
    public:  
        int id;  
        int cost;  
};
```

1
2 1
3 1
4 1
5
6
7 is 0

```
class rt-node {  
    public:
```

```
        bool visited; int cost; int from; int heap-pos; };
```

Vector<rt-node> Routing Table;

* Note: we don't need to construct heap.
as all cost is ∞ .

* Use STL for vector

* Don't use STL for heap.

```
class n-node {  
    public:  
        int id;  
        int weight;  
};
```

vector<vector<nnode*>>

graph

Vector<Vector<n node*>> graph.