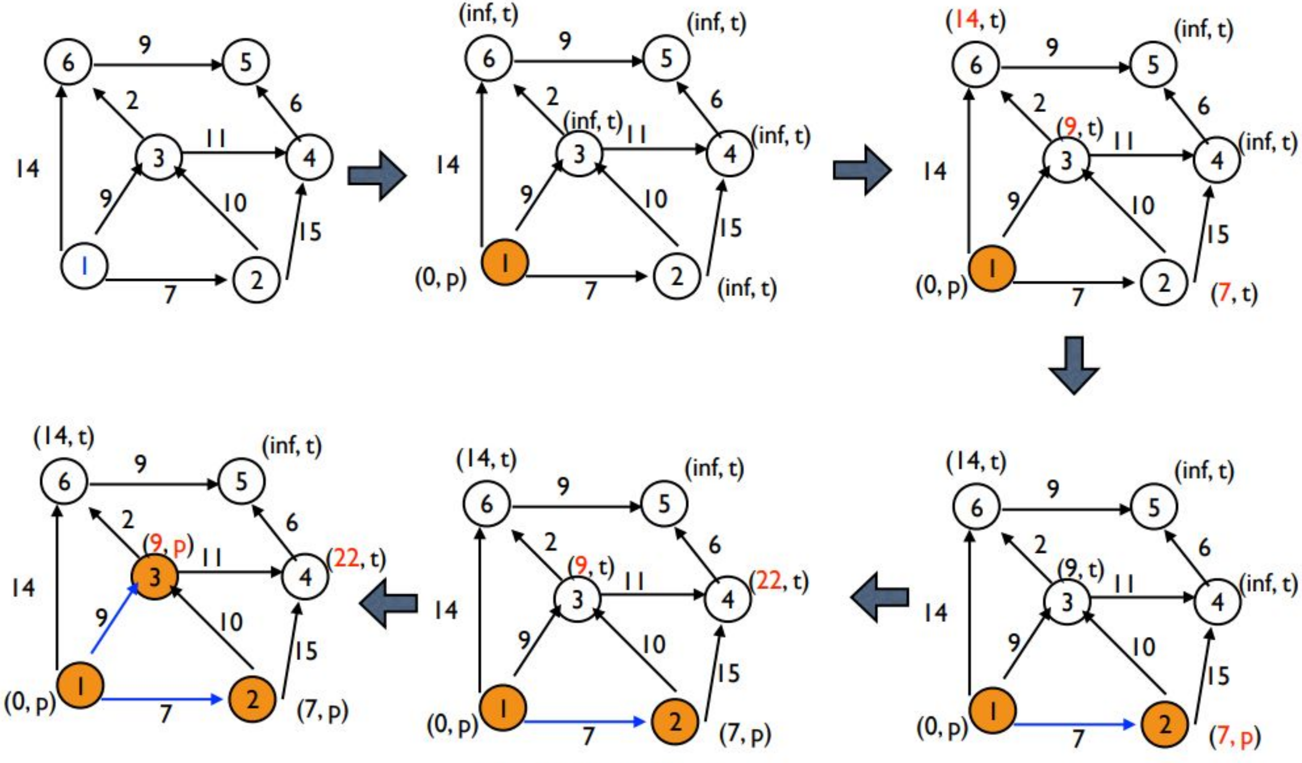
Lab 12: Dijkstra’s Algorithm

Data Structure 2023

Dijkstra’s Algorithm

- Length of a path: sum of edge weights along the path
- Finding minimum length of the path from u to v: δ(s, v)
- Given a directed graph with **non-negative** edge weights G = (V, E), and a special source vertex s ∈ V, determine the distance from the source vertex to every vertex in G
  - d[v]: shortest path from the source to v
  - pred[v]: previous vertex of v in the path
- each node is one of the status, permanent or temporary
  - the status of a node is permanent if its distance value is equal to the shortest distance from node s
  - otherwise, the status of a node is temporary

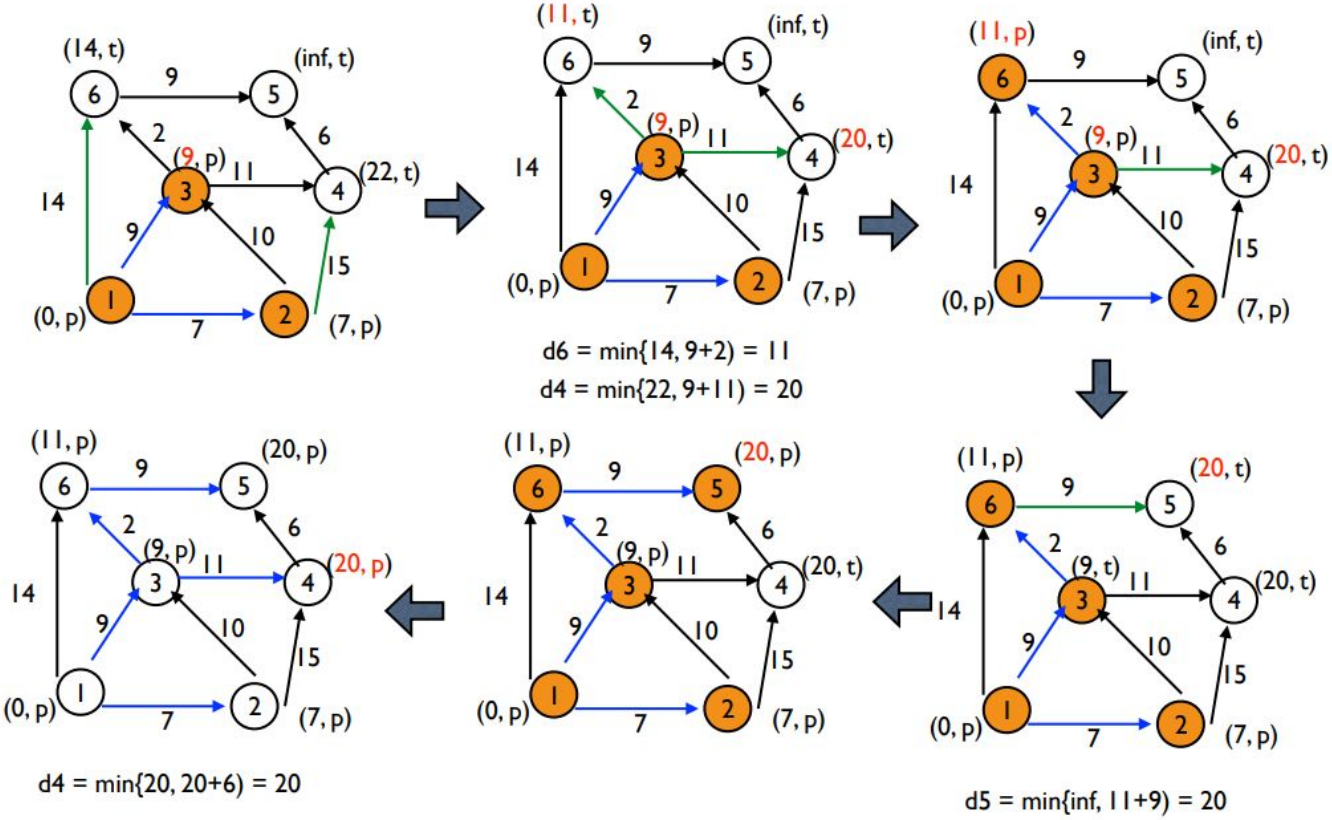
Dijkstra’s Algorithm



d3 = min{9, 7+10) = 9

d4 = min{inf, 7+15) = 22

Dijkstra’s Algorithm



d6 = min{14, 9+2) = 11

d4 = min{22, 9+11) = 20

d5 = min{inf, 11+9) = 20

d4 = min{20, 20+6) = 20

Dijkstra's Algorithm ADT

- **Graph* createGraph(int size)**
  - Create a graph with nodes.
- **void dijkstra(Grahph* g)**
  - Process dijkstra algorithm.
- **int* shortestPath(Graph* g, int dest)**
  - Return the shortest path into the given destination.
- **Heap* createMinHeap(int heapSize)**
  - Create min heap.
- **void insertToMinHeap(Heap* minHeap, int vertex, int distance)**
  - Insert a new vertex to heap.
- **Node deleteMin(Heap* minHeap)**
  - Delete the smallest distance node for calculation.
- **void deleteGraph(Graph* g) & void deleteMinHeap(Heap* minHeap)**
  - Delete a graph and min heap.

Dijkstra’s Algorithm ADT

Structure

```
typedef struct Node {
    int vertex;
    int dist;
    int prev;
}Node;
```

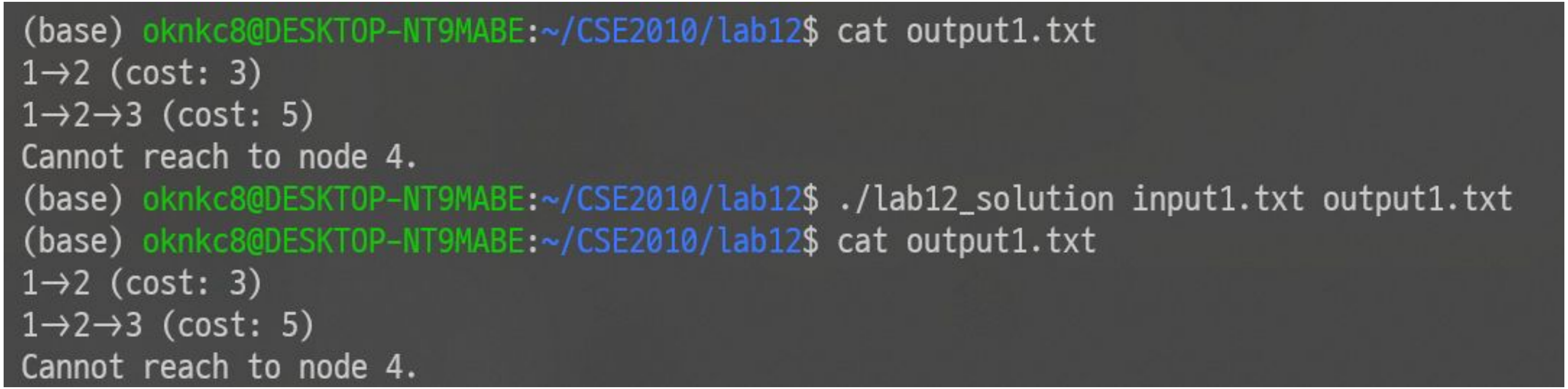
```
typedef struct Graph {
    int size;
    int** vertices;
    Node* nodes;
}Graph;
```

```
typedef struct Heap {
    int Capacity;
    int Size;
    Node* Element;
}Heap;
```

Function

```
Graph* createGraph(int size);
void deleteGraph(Graph* g);
void dijkstra(Graph* g);
int* shortestPath(Graph* g, int dest);
Heap* createMinHeap(int heapSize);
void deleteMinHeap(Heap* minHeap);
void insertToMinHeap(Heap* minHeap,
                     int vertex, int distance);
Node deleteMin(Heap* minHeap);
```

Input & Output Example



```
(base) oknkc8@DESKTOP-NT9MABE:~/CSE2010/lab12$ cat output1.txt
1→2 (cost: 3)
1→2→3 (cost: 5)
Cannot reach to node 4.
(base) oknkc8@DESKTOP-NT9MABE:~/CSE2010/lab12$ ./lab12_solution input1.txt output1.txt
(base) oknkc8@DESKTOP-NT9MABE:~/CSE2010/lab12$ cat output1.txt
1→2 (cost: 3)
1→2→3 (cost: 5)
Cannot reach to node 4.
```

Assignment

- Due
  - **~ 2023.06.07(수) 23:59**
  - Last Commit 기준

- 자세한 내용은 과제 명세 PDF 파일 참고