레포트#4

자료구조(SW) 2분반
소프트웨어학과
32170578 김산

<코드>

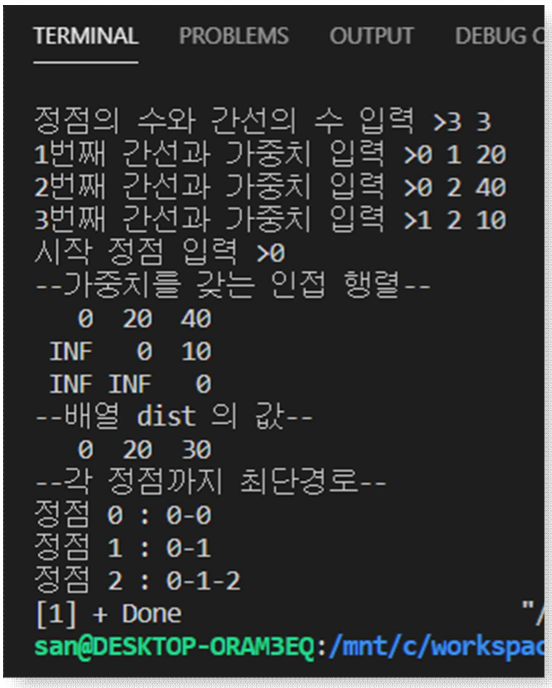
```
#include <iostream >
#include <stdbool.h >
#define INF 9999
using namespace std;
class Graph{
private:
    int **length;
    int *dist;
    int *path;
    bool *s;
    int n;
public:
    Graph(const int vertices =0): n(vertices){
        length =new int *[n];
        path =new int[n];
        for(int i =0; i < n; i ++) length[i] =new int[n];
        fill(&length[0][0],&length[n -1][n],INF);
        for (int i =0; i < n; i ++) length[i][i] =0;
    }
    void ShortestPath(const int);
    int choose(const int);
    void insert(const int, const int, const int);
    void print_adj(const int);
    void print_dist();
    void print_ShortestPath(const int,const int);
};
void Graph::ShortestPath(const int v){
    for (int i =0; i < n; i ++){ s[i] =false; dist[i] = length[v][i];} // 초기화
    s[v] =true;
    dist[v] =0;
    for (int i =0; i < n -2; i ++){
        int u = choose(n);

        s[u] =true;
        for (int w =0; w < n; w ++){
            if(!s[w]){
                if(dist[u] + length[u][w] < dist[w]){
                    dist[w] = dist[u] + length[u][w];
                    path[w] = u;
                }
            }
        }
    }
}
int Graph::choose(const int n){
    int min = INF;
    int min_index =0;
    for (int w =0; w < n; w ++)
    {
        if(s[w] ==false && dist[w] < min){
            min_index = w;
```

```
                min = dist[min_index];
            }
        }
        return min_index;
}
void Graph::insert(const int i, const int j, const int cost){
        length[i][j] = cost;
}
void Graph::print_adj(const int start){
        for (int i =0; i < n; i ++)
        {
            for (int j =0; j < n; j ++)
            {
                cout.width(4);
                if(length[i][j] == INF) cout <<"INF";
                else cout << length[i][j];
            }
            cout <<endl;
        }
}
void Graph::print_dist(){
        for (int i =0; i < n; i ++)
        {
            cout.width(4);
            cout << dist[i];
        }
        cout <<endl;
}
void Graph::print_ShortestPath(const int start, const int m){
        for (int i =0; i < n; i ++)
        {
            int last =0;
            int *s_path =new int[m];

            s_path[0] = i;
            while (path[s_path[last]] !=0)
            {
                s_path[last+1] = path[s_path[last]];
                last++;
            }
            cout <<"정점 "<< i <<" : "<< start;
            for (int j = last; j >=0; j --)
            {
                cout <<"-"<< s_path[j];
            }
            cout <<endl;
            delete s_path;
        }


}
int main(){
        int n, m;
        int i, j, cost;
        int start;
        cout <<"정점의 수와 간선의 수 입력 >";
        cin >> n >> m;
        Graph g(n);
```

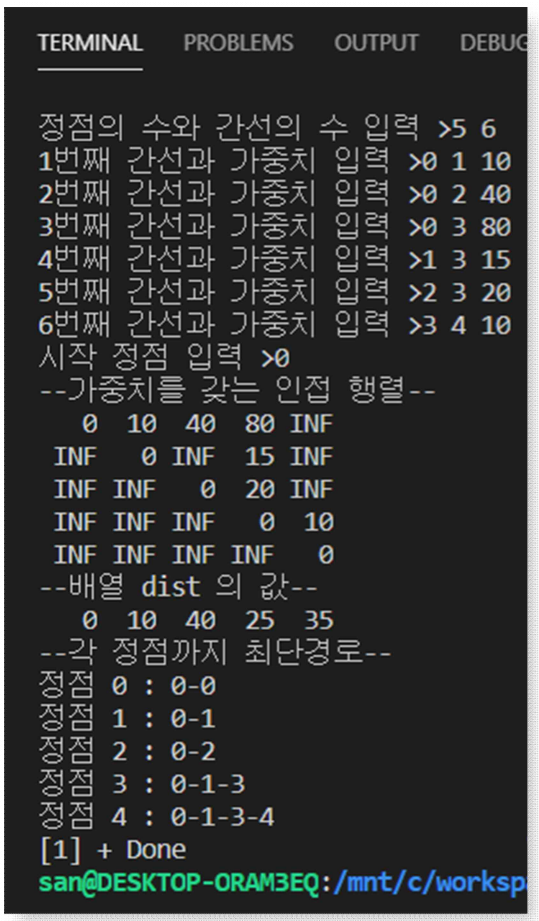
```
    for (int a =1; a <= m; a ++)
    {
        cout << a <<"번째 간선과 가중치 입력 >";
        cin >> i >> j >> cost;
        g.insert(i, j, cost);
    }
    cout <<"시작 정점 입력 >";
    cin >> start;
    g.ShortestPath(start);
    cout <<"--가중치를 갖는 인접 행렬--"<<endl;
    g.print_adj(start);

    cout <<"--배열 dist 의 값--"<<endl;
    g.print_dist();
    cout <<"--각 정점까지 최단경로--"<<endl;
    g.print_ShortestPath(start,m);
}
```

<실행화면>

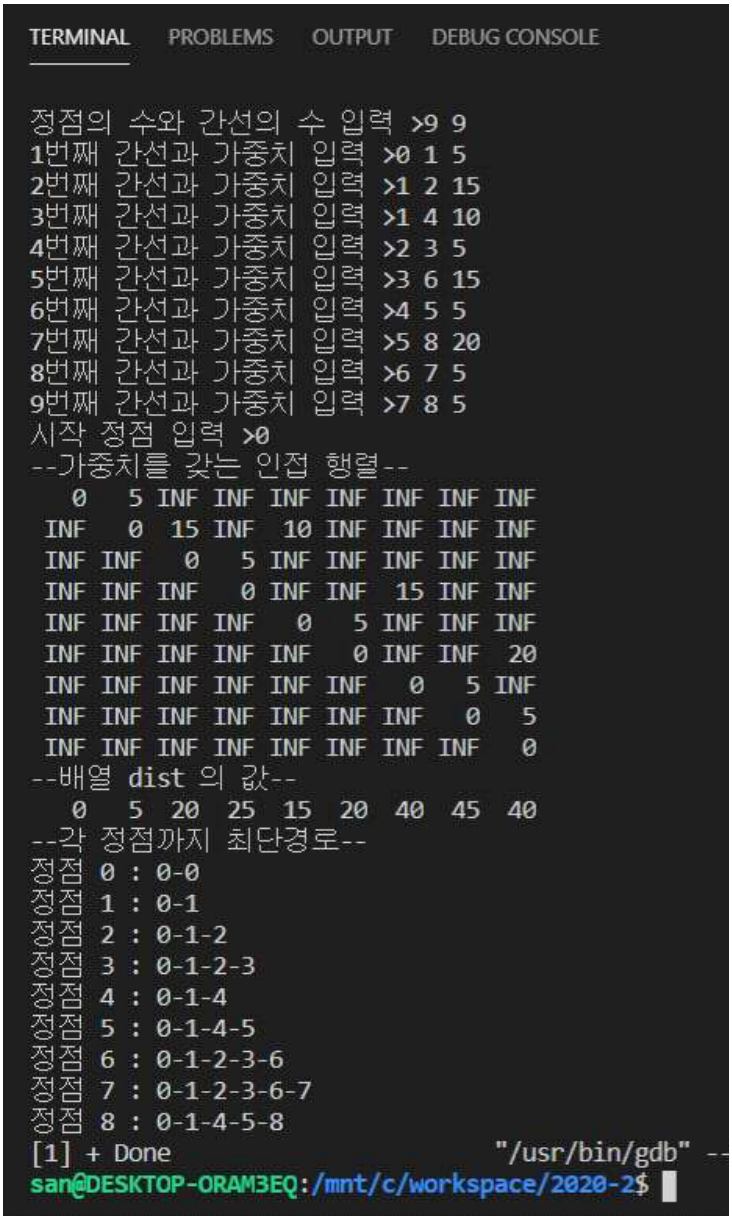
Case 1



Case 2



Case 3


TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE
정점의 수와 간선의 수 입력 >9 9
1번째 간선과 가중치 입력 >0 1 5
2번째 간선과 가중치 입력 >1 2 15
3번째 간선과 가중치 입력 >1 4 10
4번째 간선과 가중치 입력 >2 3 5
5번째 간선과 가중치 입력 >3 6 15
6번째 간선과 가중치 입력 >4 5 5
7번째 간선과 가중치 입력 >5 8 20
8번째 간선과 가중치 입력 >6 7 5
9번째 간선과 가중치 입력 >7 8 5
시작 정점 입력 >0
--가중치를 갖는 인접 행렬--
0 5 INF INF INF INF INF INF INF
INF 0 15 INF 10 INF INF INF INF
INF INF 0 5 INF INF INF INF INF
INF INF INF 0 INF INF 15 INF INF
INF INF INF INF 0 5 INF INF INF
INF INF INF INF INF 0 INF INF 20
INF INF INF INF INF INF 0 5 INF
INF INF INF INF INF INF INF 0 5
INF INF INF INF INF INF INF INF 0
--배열 dist 의 값--
0 5 20 25 15 20 40 45 40
--각 정점까지 최단경로--
정점 0 : 0-0
정점 1 : 0-1
정점 2 : 0-1-2
정점 3 : 0-1-2-3
정점 4 : 0-1-4
정점 5 : 0-1-4-5
정점 6 : 0-1-2-3-6
정점 7 : 0-1-2-3-6-7
정점 8 : 0-1-4-5-8
[1] + Done "/usr/bin/gdb" --
san@DESKTOP-ORAM3EQ:/mnt/c/workspace/2020-2$