图论模板题

DFS (dfs.cpp)

限制：1 秒　　128MB

题目描述

输入一个点数为`n`，边数为`m`的**有向图**，**并且从源点s出发**，按照点编号的升序进行**DFS(深度优先搜索)**，输出遍历顺序。

输入

第一行：`n,m,s`

第2至`m+1`行：每一行按照`u,v`格式输入输入一条边。`u`为这条边的**起始点**，`v`为这条边的**终止点**。

输出

第一行：按点的遍历顺序输出点的编号。

样例

样例一

输入

```
5 5 1
1 2
1 3
1 5
2 4
3 5
```

输出

```
1 2 4 3 5
```

数据规模

0<`n`<5000

0<`m`<100000

tip:你应该使用更少的内存空间，而不是90M以上。

伪代码

```
dfs(cur)
    cnt+1
    if cnt=n stop
    for i:1->n
        if cur-j连接 and j未遍历
            dfs(i)
```

BFS (bfs.cpp)

限制：1 秒　　128MB

题目描述

输入一个点数为`n`，边数为`m`的**有向图**，**并且从源点s出发**，按照点编号的升序进行**BFS(宽度优先搜索)**，输出遍历顺序。

输入

第一行：`n,m,s`。

第2至`m+1`行：每一行按照`u,v`格式输入输入一条边。`u`为这条边的**起始点**，`v`为这条边的**终止点**。

输出

第一行：按点的遍历顺序输出点的编号。

样例

样例一

输入

```
5 5 1
1 2
1 3
1 5
2 4
3 5
```

输出

```
1 2 3 5 4
```

数据规模

0<`n`<5000

0<`m`<100000

伪代码

```
que.push s

while not que.empty
    cur=que.front
    for i:1->n
        if cur-i连接 i未遍历
            que.push i
    que.pop
```

Floyd Warshall (floyd.cpp)

限制：1 秒　　4MB

题目描述

输入一个点数为`n`，边数为`m`的**有向图**。输出**每两个点之间的最短路径**。

输入

第一行：`n,m`。

第2至`m+1`行：每一行按照`u,v,w`格式输入输入一条边。`u`为这条边的**起始点**，`v`为这条边的**终止点**，`w`为这条边的**边权**。

输出

共`n`行`n`列：

第`i`行`j`列表示：点`i`到点`j`的**最短路径**，若`i`到`j`不连通，则输出`INF`。

样例

样例一

输入

```
4 8
1 2 2
1 3 6
1 4 4
2 3 3
3 1 7
3 4 1
4 1 5
4 3 12
```

输出

```
0 2 5 4
9 0 3 4
6 8 0 1
5 7 10 0
```

数据规模

0<`n`<430

0<`m`<8600

500<=`w`<=1000

伪代码

```
for k:1->n
    for i:1->n
        for j:1->n
            e[i][j]=min(e[i][k]+e[k][j],e[i][j])
```

Dijkstra normal (dij.cpp)

限制：1 秒　　128MB

题目描述

输入一个点数为`n`，边数为`m`的**有向图**。输出**源点到其余个点的最短路径**。

输入

第一行：`n,m`

第2至`m+1`行：每一行按照`u,v,w`格式输入输入一条边。u为这条边的**起始点**，u为这条边的**终止点**，w为这条边的**边权**。

输出

第一行：共`n`个元素，第`i`个元素表示：**源点到点i的最短路径**，若源点到`i`不连通，则输出`INF`。

样例

样例一

输入

```
4 8
1 2 2
1 3 6
1 4 4
2 3 3
3 1 7
3 4 1
4 1 5
4 3 12
```

输出

```
0 2 5 4
```

数据规模

0<`n`<5000

0<`m`<100000

500<=`w`<=1000

伪代码

```
init dis

for i:1->n-1
    min=find(dis)
    book u
    for j:1->n
        if !book j
            dis[j]=min(dis[u]+S(u->j),dis[j])
```

Dijkstra&heap (dij_heap.cpp)

限制：1 秒　　8MB

题目描述

输入一个点数为`n`，边数为`m`的有**向图**。输出**源点到其余个点的最短路径**。

输入

第一行：`n,m`

第2至m+1行：每一行按照`u,v,w`格式输入输入一条边。u为这条边的**起始点**，u为这条边的**终止点**，w为这条边的**边权**。

输出

第一行：共`n`个元素，第`i`个元素表示：**源点到点`i`的最短路径**，若源点到`i`不连通，则输出`INF`。

样例

样例一

输入

```
4 8
1 2 2
1 3 6
1 4 4
2 3 3
3 1 7
3 4 1
4 1 5
4 3 12
```

输出

```
0 2 5 4
```

数据规模

0<`n`<100000

0<`m`<200000

500<=`w`<=1000

伪代码

```
init dis

for i:1->n-1
    min=find(dis)
    book u
    for j:1->n
        if !book j
            dis[j]=min(dis[u]+S(u->j),dis[j])
```

SPFA

限制：1 秒　　8MB

题目描述

输入一个点数为`n`，边数为`m`的有**向图**。输出**源点到其余个点的最短路径。**

输入

第一行：`n,m`

第2至m+1行：每一行按照`u,v,w`格式输入输入一条边。u为这条边的**起始点**，u为这条边的**终止点**，w为这条边的**边权。**

输出

第一行：共`n`个元素，第`i`个元素表示：**源点到点`i`的最短路径**，若源点到`i`不连通，则输出`INF`。

样例

样例一

输入

```
4 8
1 2 2
1 3 6
1 4 4
2 3 3
3 1 7
3 4 1
4 1 5
4 3 12
```

输出

```
0 2 5 4
```

数据规模

0<`n`<100000

0<`m`<200000

-500<=`w`<=1000

伪代码

```
init dis,cnt,book
que.push s

while not q.empty
    u=q.front
    q.pop
    for q->v
        if dis[u]<dis[v]+S(u->v)
            dis[u]=dis[v]+S(u->v)
            if not book v
                cnt[v]+1
                if cnt[v]==n
                    NO ANSWER
                q.push v
                book v
```