F 上班 I

21371300 柳政尧

题目描述

观赛完运动会之后，Zhoues 需要立马回去钢条厂上班，由于北航到钢条厂的路径有很多条，他希望选择一条最短的！

形式化来说，给一个 n 个点 m 条边的带权无向图，求点 s 到点 t 的最短路长度。

输入格式

第一行四个正整数 n, m, s, t（$1 \leq n \leq 2.5 \times 10^3$，$1 \leq m \leq 7 \times 10^3$，$1 \leq s, t \leq n$），含义如题目所示。

接下来 m 行，每行三个正整数 u_i, v_i, w_i（$1 \leq u_i, v_i \leq n$，$1 \leq w_i \leq 10^3$），表示第 i 条边 (u_i, v_i) 的权值为 w_i。

图中有可能存在重边和自环。

删掉自环！

Dijkstra!

代码实现

采用邻接表存图的典型模板。

```
16  struct Edge
17  {
18      int v;
19      int w;
20      int next;
21  };
22
23  int     head[MAX_VERTEX_NUM];
24  Edge    edge[MAX_EDGE_NUM * 2];
25  bool    flag[MAX_VERTEX_NUM];
26  int     dist[MAX_VERTEX_NUM];
27  int     edgestamp;
28
29  void addEdge(int u, int v, int w);
30  void _addSingleEdge(int u, int v, int w);
31  int dijkstra(int src, int dest);
```

代码实现

由于Dijkstra采用贪心的思想，因此自环是一定不会被选择的。

```
51  void addEdge(int u, int v, int w)
52  {
53      if (u != v)
54      {
55          _addSingleEdge(u, v, w);
56          _addSingleEdge(v, u, w);
57      }
58  }
59
60  void _addSingleEdge(int u, int v, int w)
61  {
62      edgestamp++;
63      edge[edgestamp].next = head[u];
64      edge[edgestamp].v = v;
65      edge[edgestamp].w = w;
66      head[u] = edgestamp;
67  }
```

代码实现

Dijkstra 算法也是模板，可以使用 C++ STL 中的 priority_queue 简化实现，包含在 <queue> 头文件中。

```cpp
int dijkstra(int src, int dest)
{
    std::priority_queue<
        std::pair<int, int>,
        std::vector<std::pair<int, int>>,
        std::greater<std::pair<int, int>>> heap;
    int u, v, w;

    memset(dist, 0x5f, sizeof(dist));
    dist[src] = 0;
    heap.push(std::make_pair(dist[src], src));
    while (!heap.empty())
    {
        u = heap.top().second;
        heap.pop();

        if (flag[u])
            continue;
        flag[u] = true;

        for (int i = head[u]; i != 0; i = edge[i].next)
        {
            v = edge[i].v;
            w = edge[i].w;

            if (dist[v] > dist[u] + w)
            {
                dist[v] = dist[u] + w;
                if (!flag[v])
                    heap.push(std::make_pair(dist[v], v));
            }
        }
    }

    return dist[dest];
}
```

T H A N K S !

21371300 柳政尧