

有向图,  $s$  到  $t$  的第  $k$  短路。

```

1  int n , m , s , t , k ;
2  vector<pii> g1[N] , g2[N] ;
3  struct Dij
4  {
5      int dis[N] ;
6      priority_queue<pii , vector<pii> , greater<pii> > q ;
7      void init()
8      {
9          memset(dis , inf , sizeof(dis)) ;
10     }
11     void dijkstra(int s)
12     {
13         dis[s] = 0 ;
14         q.push(make_pair(0 , s)) ;
15         while(!q.empty())
16         {
17             pii p = q.top() ;
18             q.pop() ;
19             int u = p.second ;
20             if(p.first != dis[u]) continue ; //优化, 不用旧值更新。
21             for(vector<pii>::iterator x = g2[u].begin() ; x != g2[u].end() ; x ++ )
22                 //for(auto x : g2[u])
23                 {
24                     int v = (*x).fi , w = (*x).se ;
25                     if(dis[v] > dis[u] + w)
26                     {
27                         dis[v] = dis[u] + w ;
28                         q.push(make_pair(dis[v] , v)) ;
29                     }
30                 }
31         }
32     }
33 } dij ;
34 struct node
35 {
36     int u , g , h ;
37     bool operator < (const node &s) const
38     {
39         return g + h > s.g + s.h ;
40     }
41 } ;
42 priority_queue<node> q ;
43 int num[N] ;
44 int astar()
45 {
46     mem0(num) ;
47     while(!q.empty()) q.pop() ;
48     q.push(node{s , 0 , dij.dis[s]}) ;
49     while(!q.empty())
50     {
51         int u = q.top().u , g = q.top().g ;

```

```
52     q.pop() ;
53     num[u] ++ ;
54     if(num[u] == k && u == t) return g ;
55     if(num[u] > k) continue ;
56     for(vector<pii>::iterator x = g1[u].begin() ; x != g1[u].end() ; x ++ )
57         //for(auto x : g1[u])
58     {
59         int v = (*x).fi , w = (*x).se ;
60         q.push(node{v , g + w , dij.dis[v]}) ;
61     }
62 }
63 return -1 ;
64 }
65 int main()
66 {
67     rr(n , m) ;
68     rep(i , 1 , m)
69     {
70         int u , v , w ;
71         rr(u , v) , r(w) ;
72         g1[u].pb({v , w}) ;
73         g2[v].pb({u , w}) ;
74     }
75     rr(s , t) , r(k) ;
76     if(s == t) k ++ ;
77     dij.init() ;
78     dij.dijkstra(t) ;
79     we(aster()) ;
80     return 0 ;
81 }
```