C++算法

快快编程1088,1090

BFS最短路计数

cnt[v]代表从源点到v的最短路径有几条

BFS过程中,所有边长均为1
当访问从u到v的边时

如果v曾经访问过,且d[v]==d[u]+1

- d[v]不变
- cnt[v]=(cnt[v]+cnt[u])%MOD

如果v之前未访问过

- vst[v]=1, d[v]=d[u]+1
- cnt[v]=cnt[u]

```
32      cin>>n>>m;
33      for(int k=1;k<=m;k++){
34          int a,b;
35          cin>>a>>b;
36          to[a].push_back(b);
37          to[b].push_back(a);
38      }
39      bfs();
40      for(int u=1;u<n;u++)cout<<cnt[u]<<" ";
41      cout<<cnt[n]<<endl;
```

最短路计数

快快1088

```
9  void bfs(){
10     queue<int> q;
11     vst[1]=1;
12     d[1]=0;
13     cnt[1]=1;
14     q.push(1);
15     while(!q.empty()){
16         int u=q.front(); q.pop();
17         for(int i=0;i<to[u].size();++i){
18             int v=to[u][i];
19             if(!vst[v]){
20                 vst[v]=1;
21                 d[v]=d[u]+1;
22                 
23                 q.push(v);
24             }else if(d[v]==d[u]+1)
25                 cnt[v]=
26         }
27     }
28 }
```

快快1090

Dijkstra最短路计数

cnt[v]代表从源点到v的最短路径有几条

Dijkstra过程中
对于一条边: 从u到v长度为w

如果d[v]==d[u]+w

- d[v]不变
- cnt[v]=(cnt[v]+cnt[u])%MOD

如果d[v]>d[u]+w

- d[v]=d[u]+w
- cnt[v]=cnt[u]

```
42      cin>>n>>m;
43      for(int k=1,a,b,c;k<=m;k++){
44          cin>>a>>b>>c;
45          to[a].push_back(b);
46          to[b].push_back(a);
47          w[a].push_back(c);
48          w[b].push_back(c);
49      }
50      Dijkstra();
51      for(int u=1;u<n;u++) cout<<cnt[u]<<" ";
52      cout<<cnt[n]<<endl;
```

最短路计数

快快1090

```
15  void Dijkstra(){
16      fill(d,d+n+9,INF);
17      priority_queue<Node> q;
18      d[1]=0;
19      cnt[1]=1;
20      q.push((Node){1,0});
21      while(!q.empty()){
22          int u=q.top().u; q.pop();
23          
24          ok[u]=1;
25          for(int i=0;i<to[u].size();++i){
26              int v=to[u][i];
27              if(ok[v])continue;
28              int cost=d[u]+w[u][i];
29              if(d[v]==cost)
30                  cnt[v]=(cnt[v]+cnt[u])%MOD;
31              else if(d[v]>cost){
32                  d[v]=cost;
33                  
34                  q.push((Node){v,d[v]});
35              }
36          }
37      }
38  }
```

现场挑战
快快编程1092

社交网络

对于每个节点v，求出所有C(s,t,v)
s到t的最短路有几条经过v

对于任意两个节点s,t,求出所有C(s,t)
s到t的最短路有几条

以上2个问题先求解哪个?

NOI2007

社交网络

对于任意两个节点s,t,求出所有C(s,t)
s到t的最短路有几条

多源多汇用Floyd-Warshall框架

d[i][j][k]: 只借助1到k，从i到j的最短路

cnt[i][j]: 从i到j的最短路共几条

```
for(ll k=1;k<=n;k++)
    for(ll i=1;i<=n;i++)
        for(ll j=1;j<=n;j++){
            ll cost=d[i][k]+d[k][j];
            if(d[i][j]==cost)cnt[i][j]+=
            else if(d[i][j]>cost)
                d[i][j]=cost,cnt[i][j]=
        }
```

NOI2007

社交网络

```
for(ll v=1;v<=n;v++)
    for(ll s=1;s<=n;s++)if(s!=v){
        for(ll t=1;t<=n;t++)if(t!=v&&t!=s){
            if(d[s][t]==d[s][v]+d[v][t])
                ans[v]+=
        }
    }
cout<<fixed<<setprecision(3);
for(ll v=1;v<=n;v++) cout<<ans[v]<<endl;
```

现场挑战
快快编程1087

严格次短路

错误算法

先求出最短路径p
删除p中某条边后，再求最短路
其中最短的就是次短路

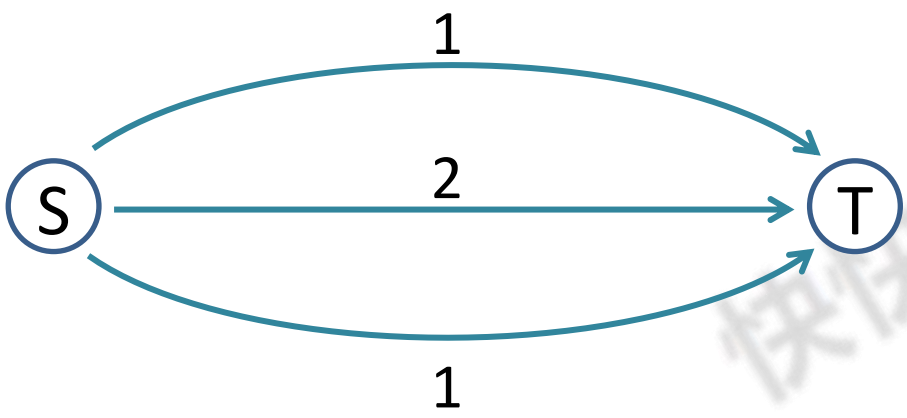
请描述一个反例

严格次短路

错误算法

先求出最短路径p
删除p中某条边后，再求最短路
其中最短的就是次短路

反例

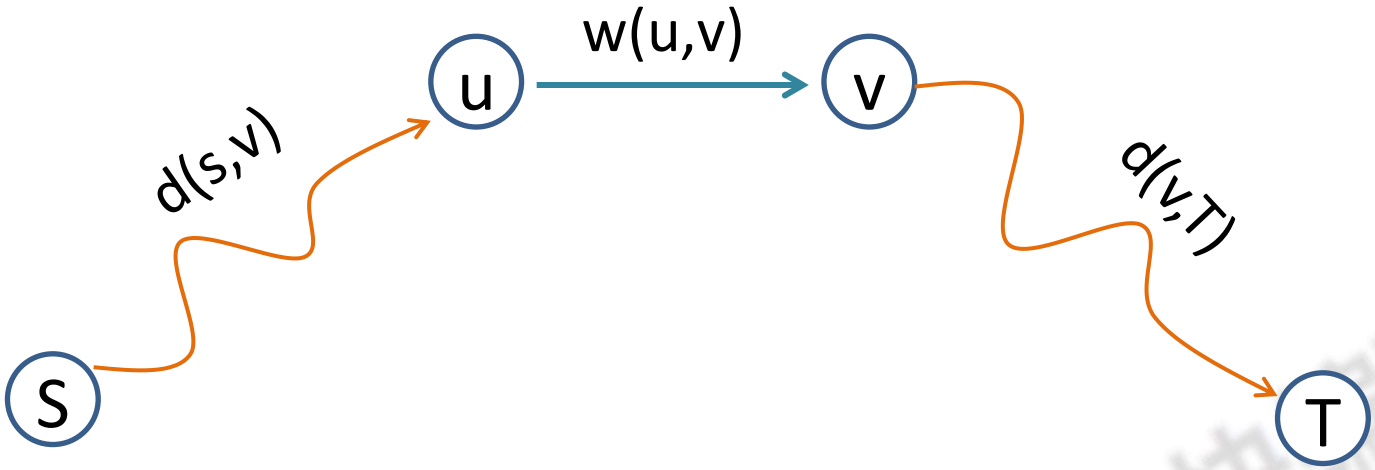


严格次短路

单源单汇

路径分段法

枚举分割段



严格次短路

单源单汇

路径分段法

枚举分割段

跑两次Dijkstra堆优化
dS[u]: 从源点S到u最短路长度
dT[v]: 从v到汇点T最短路长度

枚举每条边(u,v)
计算dS[u]+w(u,v)+dT[v]
可能拼出次短路

求证：以上方法不会遗漏次短路

严格次短路

错误算法

路径分段法

枚举分割点

跑两次Dijkstra堆优化
dS[u]: 从源点S到u最短路长度
dT[v]: 从v到汇点T最短路长度

枚举每个节点u
计算dS[u]+dT[u]
尝试拼出次短路

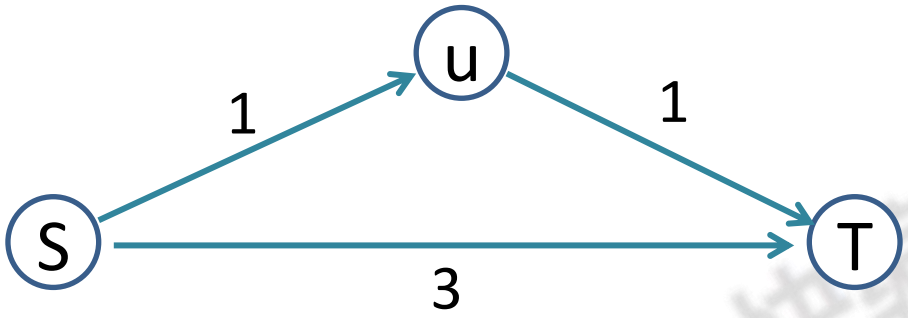
请给出最简反例

严格次短路

错误算法

- 路径分段法
- 枚举分割点

反例



```cpp
void Dijkstra(int s,int*d){
    fill(ok,ok+n+9,0);
    fill(d,d+n+9,INF);
    priority_queue<Node> q;
    d[s]=0;
    q.push((Node){s,0});
    while(!q.empty()){
        int u=q.top().u; q.pop();
        if(ok[u])continue;
        ok[u]=1;
        for(int i=0;i<to[u].size();++i){
            int v=to[u][i];
            if(ok[v])continue;
            int cost=d[u]+w[u][i];
            if(d[v]<=cost)continue;
            d[v]=cost;
            q.push((Node){v,d[v]});
        }
    }
}
```

能否删除这句

```
45      Dijkstra(1,d1);
46      Dijkstra(n,dn);
47      int ans=INF;
48      for(int u=1;u<=n;u++)
49          for(int i=0;i<to[u].size();++i){
50              int v=to[u][i];
51              int cost=d1[u]+w[u][i]+dn[v];
52              if(                          )ans=cost;
53          }
54      cout<<ans<<endl;
```

快快编程作业

1088

1090

1092

1087