

太戈编程
etiger.vip

信奥算法

最短路问题

单源多汇SSSP

single source shortest path

Dijkstra算法

火箭1

有 n 个星球编号1到 n 。星球间有 m 个直飞火箭航班，单向飞行。求1号星球到各星球最少转几次航班？如果无法到达用-1表示。

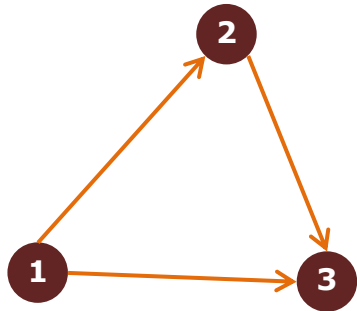
输入第一行为正整数 n 和 m ， $n \leq 100, m \leq 2000$ 。接着 m 行每行两个数 u 和 v 代表从 u 可以直飞到 v 。

输入样例

3 3
1 2
2 3
1 3

输出样例

0 1 1

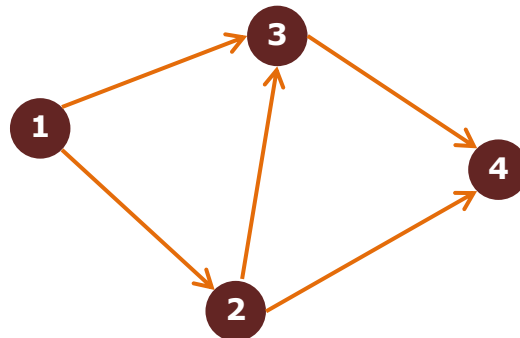


输入样例

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

输出样例

0 1 1 2



图论建模

每个节点代表什么？

每条边代表什么？

有向边还是无向边？

有权边还是无权边？

太戈编程
www.etiger.vip

```
4 vector<int> to[N];
5 bool vst[N];
6 int n,m,d[N];
7 void bfs(){
27 int main(){
28     cin>>n>>m;
29     for(int i=1;i<=m;++i){
30         int u,v;
31         cin>>u>>v;
32         to[u].push_back(v);
33     }
34     bfs();
35     return 0;
36 }
```

```

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

```

vst[u]代表u是否访问过

d[u]代表从起点到u的最短路长度

无权图最短路BFS

贪心
思维

从起点一层一层访问节点
呈现出扩散形态

每个节点 u 在第一次被访问时
就能确定 $d[u]$ 从起点到 u 的最短路

每个节点最多访问1次
每条边最多访问1次

时间复杂度

$O(\text{点数} + \text{边数})$

非负权图最短路

Dijkstra

火箭2

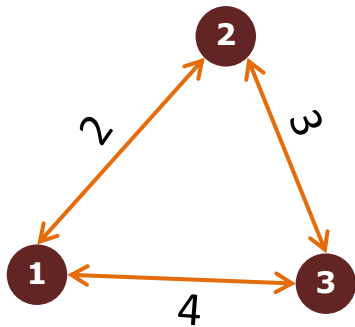
有 n 个星球编号1到 n 。星球间可能有直飞火箭往返，每条火箭航线的航行时间用邻接矩阵表示，如果两个星球间没有直飞用-1表示。求1号星球到各星球最少时间？
保证所有星球都能从1号飞到。 $n \leq 100$

输入样例

```
3
0 2 4
2 0 3
4 3 0
```

输出样例

```
0 2 4
```

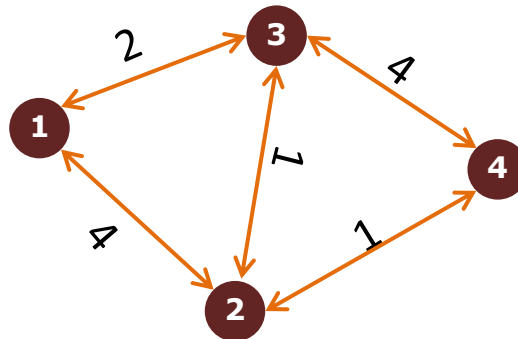


输入样例

```
4
0 4 2 -1
4 0 1 1
2 1 0 4
-1 4 1 0
```

输出样例

```
0 3 2 4
```

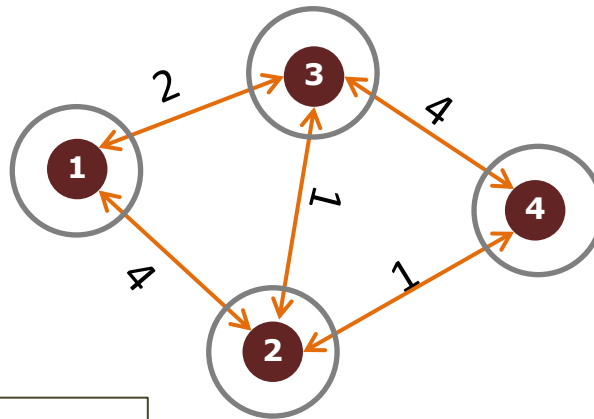


不一定是
最终结果

$d[u]$ 记录当前从起点到 u 的最短路长度

$d[1]$ 初始为0, 其他 $d[u]$ 为INF

u	1	2	3	4
d[u]	0	3	2	4



贪心法

Dijkstra

每次找没确定答案的节点里 d 值最小的
 $d[u]$ 确定为答案,再尝试更新 u 邻居的答案

编程
tiger.vip

$d[u]$ 记录当前从起点
到 u 的最短路长度

$ok[u]$ 记录
 $d[u]$ 是否为最终值

$d[1]$ 初始为0
其他 $d[u]$ 为 INF

$ok[u]$ 全部
初始为0

主循环：以下操作重复共 n 次

找不 ok 节点里 d 值最小节点 u

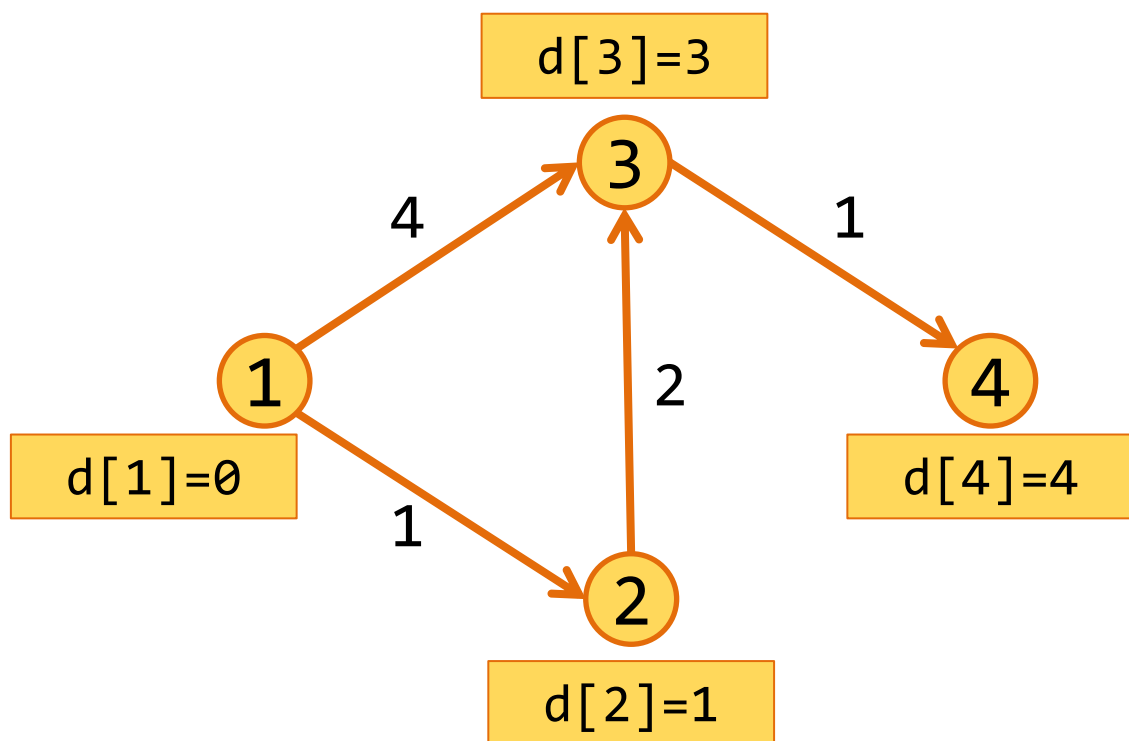
设置 $d[u]$ 值为最终值

从 u 出发找 u 的所有邻居 v

尝试更新 $d[v]$ 值

时间复杂度

$O(\text{点数}^2)$

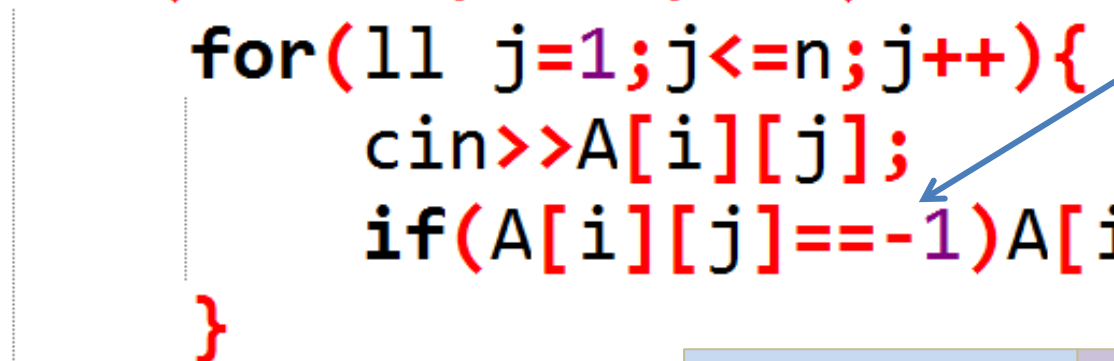


u	d[u]	ok[u]
1	0	1
2	1	1
3	3	1
4	4	1

Dijkstra
过程演示

每条边(u,v)只会访问1次
在ok[u]确定时尝试更新d[v]

```
22     cin>>n;
23     for(ll i=1;i<=n;i++)
24         for(ll j=1;j<=n;j++){
25             cin>>A[i][j];
26             if(A[i][j]==-1)A[i][j]=INF;
27         }
28     Dijkstra();
```



图的储存	邻接矩阵
------	------

A[i][j]存储i到j的边

```
8 void Dijkstra(){
9     fill(d,d+n+2,INF);
10    fill(ok,ok+n+1,0);
11    d[1]=0;
12    for(ll k=1;k<=n;k++){
13        ll u=n+1;
14        for(ll v=1;v<=n;v++)
15            if(!ok[v]&& d[v]<d[u]) u=v;
16        ok[u]=1;
17        for(ll v=1;v<=n;v++)
18            d[v]=min(d[v],d[u]+A[u][v]);
19    }
20 }
```

找不ok节点里d值最小节点u

设置d[u]值为最终值

从u出发找u的所有邻居v

尝试更新d[v]值

Dijkstra正确性

$d[u]$ 记录当前从起点
到 u 的最短路长度

$ok[u]$ 记录
 $d[u]$ 是否为最终值

$d[1]$ 初始为0
其他 $d[i]$ 为INF

$ok[u]$ 全部
初始为0

源点到第1个ok的节点 u_1 的最短路一定是 $d[u_1]$

源点到第2个ok的节点 u_2 的最短路一定是 $d[u_2]$

...

源点到第 n 个ok的节点 u_n 的最短路一定是 $d[u_n]$

Dijkstra正确性

算法核心是贪心法思维

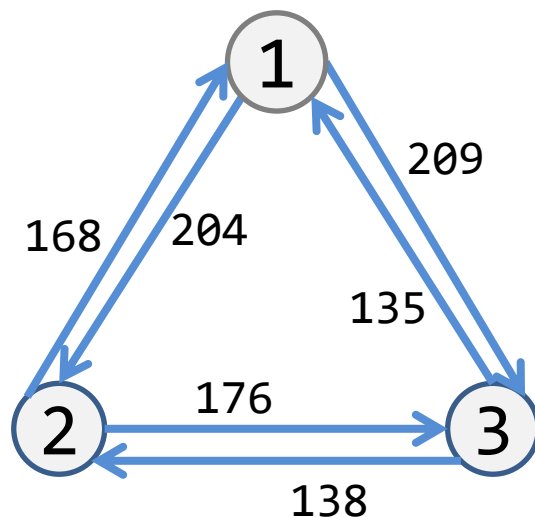
当前 $d[u]$ 标记**最小**的一定就是最终答案

因为边长都是非负数值

每次都选 $d[u]$ 标记**最小**的节点
继续更新其邻居的 d 值

638

把样例数据转换为图论信息画出来



从1号到2号路费 $(1*666+2*2) \bmod 233$ 等于204;

从2号到1号路费 $(2*666+1*1) \bmod 233$ 等于168;

从1号到3号路费 $(1*666+3*3) \bmod 233$ 等于209;

从3号到1号路费 $(3*666+1*1) \bmod 233$ 等于135;

从2号到3号路费 $(2*666+3*3) \bmod 233$ 等于 176 ;

从3号到2号路费 $(3*666+2*2) \bmod 233$ 等于138

图论 建模

把原问题已知条件转为图论信息

每个节点代表什么含义

每条边代表什么含义

边有没有方向?

共有几个节点?

n

共有几条边?

$n*(n-1)$

有权边还是无权边?

从u到v的边长计算方法?

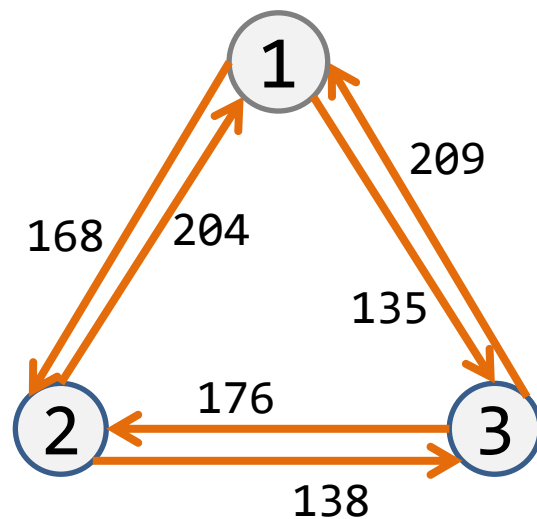
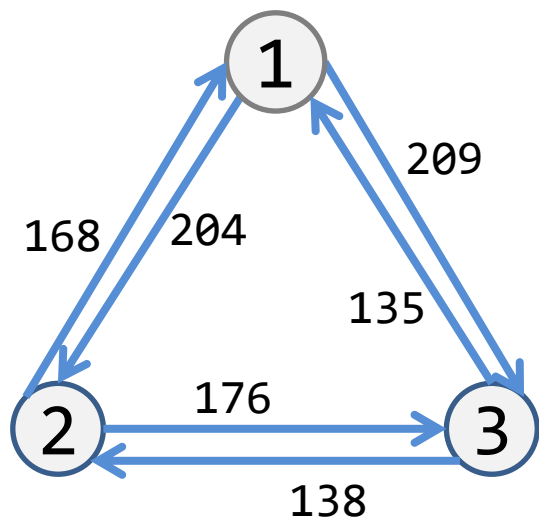
把求解的问题转为图论的问题描述

求有向非负权图多源单汇最短路

多源单汇 转换为 单源单汇

所有边反向

把样例数据转换为图论信息画出来



请同学完成程序
开头的注释分析

3分钟后
老师检查

```
1  /*姓名XXX
2  图论建模:
3  已知:
4  每个节点代表
5  每条边代表
6  边有没有方向?
7  共有几个节点?
8  共有几条边?
9
10  有权边还是无权边?
11  原图里从u到v的边长计算方法?
12
13  求解:
14  求有向非负权图多源单汇最短路?
15
16  多源单汇 转换为 单源单汇
17  所有边反向
18
19  新图里从u到v的边长计算方法?
20  */
```

```
20 int main(){
21     freopen("meeting.in", "r", stdin);
22     freopen("meeting.out", "w", stdout);
23     cin >> n;
24     for(int i=1; i<=n; i++)
25         for(int j=1; j<=n; j++)
26             A[i][j] = 
27     Dijkstra();
28     int ans=0;
29     for(int i=1; i<=n; i++) ans+=d[i];
30     cout << ans << endl;
31     return 0;
32 }
```

```
7 void Dijkstra(){
8     fill(ok, ok+n+1, 0);
9     fill(d, d+n+2, INF);
10    d[1]=0;
11    for(int k=1; k<=n; k++){
12        int now=n+1;
13        for(int j=1; j<=n; j++){
14            if( ) now=j;
15            for(int j=1; j<=n; j++){
16                d[j]=
17            }
18        }
19    }
```

1817

图论 建模

把原问题已知条件转为图论信息

每个节点代表什么含义

每条边代表什么含义

边有没有方向?

共有几个节点?

$10*9$

共有几条边?

约 $10*9*4$

有权边还是无权边?

把求解的问题转为图论的问题描述

求有向无权图单源单汇最短路

隐式图：不需要显性建立边的连接


```
1  #include<bits/stdc++.h>
2  using namespace std;
3  const int N=509;
4  struct Node{int x,y;};
5  int dx[8]={1,1,-1,-1,2,2,-2,-2};
6
7  bool vst[N][N];
8  int r1,c1,r2,c2,d[N][N];
9  int n=10,m=9;
10 void bfs(){
29 int main(){
30     freopen("horse.in","r",stdin);
31     freopen("horse.out","w",stdout);
32     cin>>r1>>c1>>r2>>c2;
33     bfs();
34     return 0;
35 }
```

```

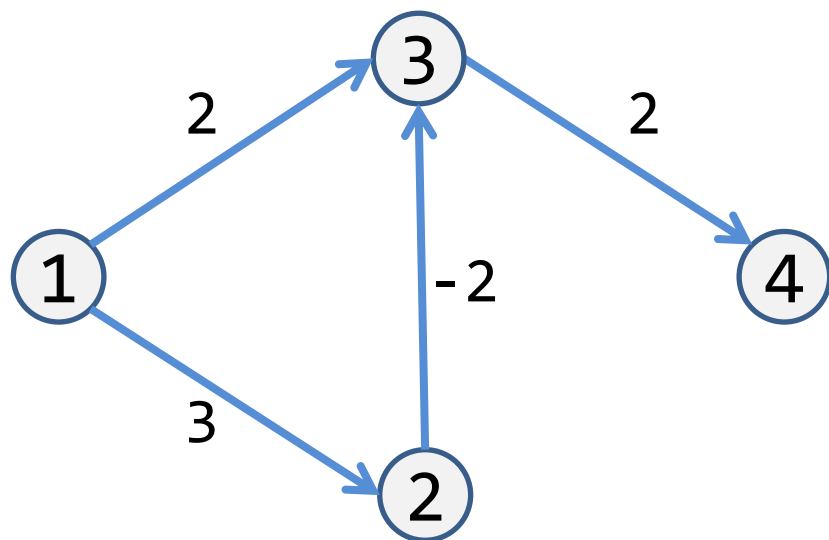
10 void bfs(){
11     queue<Node> q;
12     vst[r1][c1]=1;
13     d[r1][c1]=0;
14     q.push((Node){r1,c1});
15     while(!q.empty()){
16         Node now=q.front(); q.pop();
17         for(int k=0;k<8;k++){
18             int nx=now.x+dx[k],ny=now.y+dy[k];
19             if(nx<1||nx>n||ny<1||ny>m)continue;
20             if( )continue;
21             vst[nx][ny]= ;
22             d[nx][ny]= ;
23             q.push((Node){nx,ny});
24         }
25     }
26     if( )cout<<-1<<endl;
27     else cout<<d[r2][c2]<<endl;
28 }

```

Dijkstra正确性

假设出现负权边，
Dijkstra算法求出的最短路还正确吗

请画出一个简单的反例
包含负权边
使Dijkstra算法出错



错误答案

u	d[u]	ok[u]
1	0	1
2	3	1
3	1	1
4	4	1

Dijkstra错误地确认 $d[3]=2$ 为最终值
 $d[3]$ 获得了修改后续邻居的资格

$d[3]$ 帮助修改 $d[4]$ 一次后
 $d[4]$ 再也没有机会改正了
 虽然 $d[3]$ 可以被 $d[2]$ 改正

Dijkstra算法里
 每条边最多只会
 使用1次

如何修改算法
 使结果正确?

Dijkstra	Floyd-Warshall
点核心	点核心
$O(n^2)$, 堆优化 $O(m\log n)$	$O(n^3)$
SSSP	任意两点最短路
怕负边	不怕负边
怕负环	能判断负环
邻接矩阵	邻接矩阵
邻接表	

SSSP演示

算法可视化网址：
visualgo.net/en/sssp

Dijkstra's algorithm is the most frequently used SSSP algorithm for typical input: Directed weighted graph that has **no negative weight edge at all**

太戈编程

1817

637

638