



路径问题

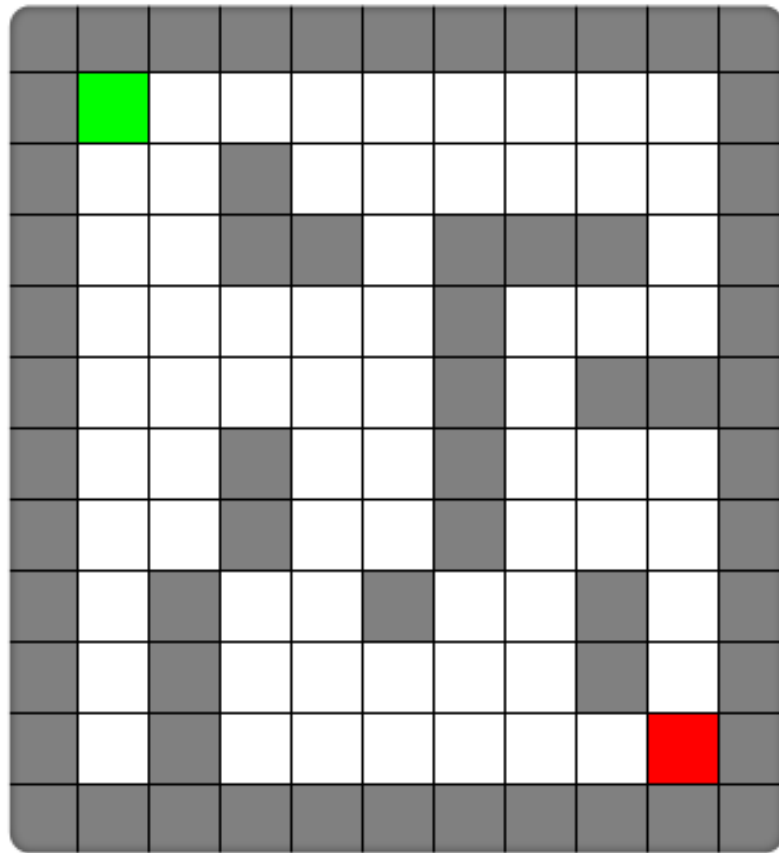


西南大学附属中学
High School Affiliated to Southwest University

灰色代表墙壁，绿色代表起点，红色代表终点，规定每次只能走一步，且只能往下或右走。求一条绿色到红色的最短路径。

DFS的解决办法：

搜索每一条路径，保存最小的步数(最优剪枝)





路径问题

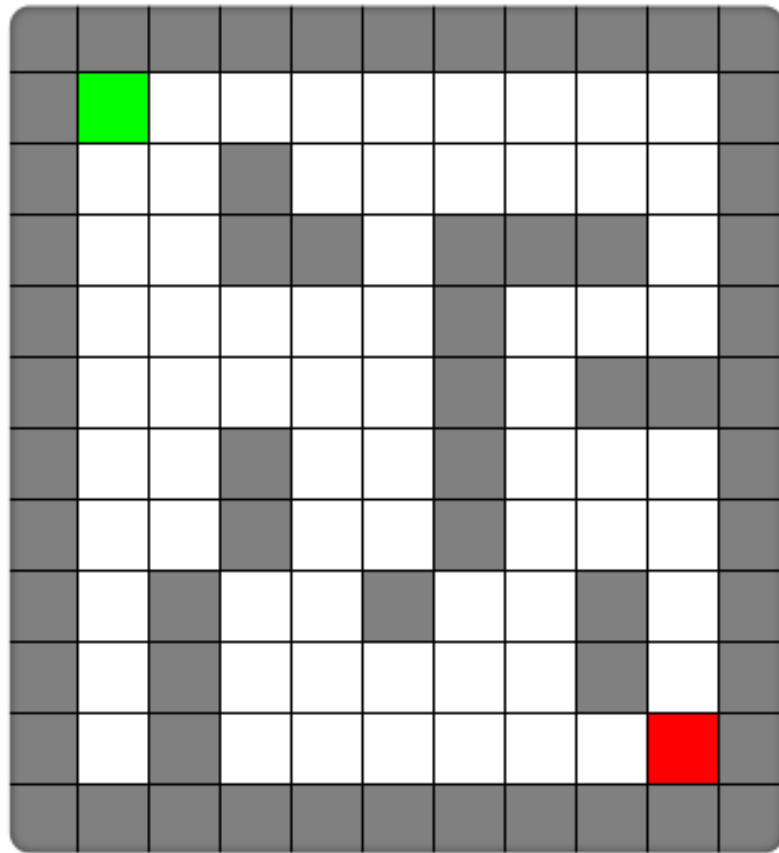


西南大学附属中学
High School Affiliated to Southwest University

灰色代表墙壁，绿色代表起点，红色代表终点，规定每次只能走一步，且只能往下或右走。求一条绿色到红色的最短路径。

问题在于：

由于不知道哪一条是最短路径，所以需要搜遍整张图

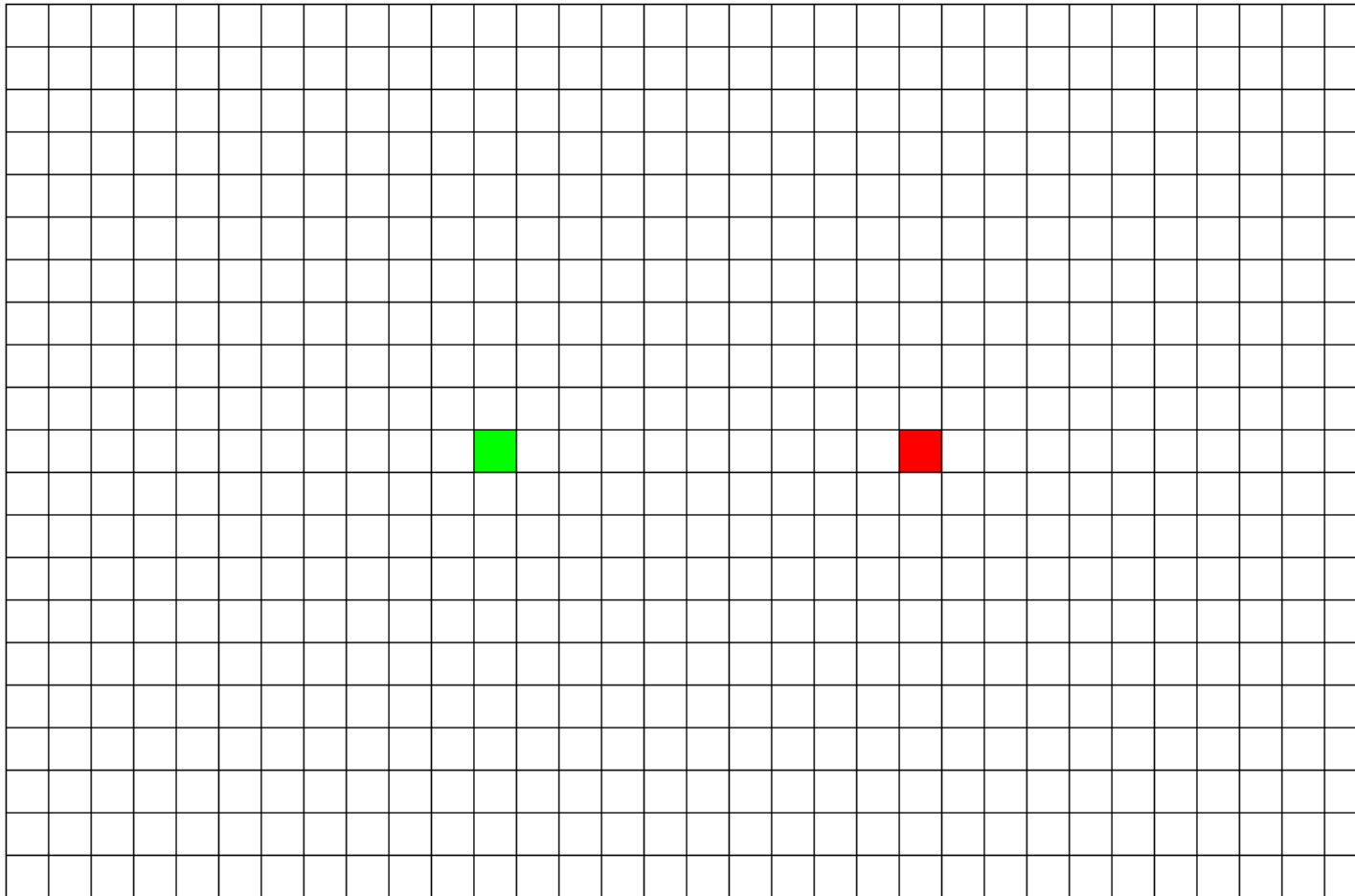




DFS极端情况(动画)



西南大学附属中学
High School Affiliated to Southwest University



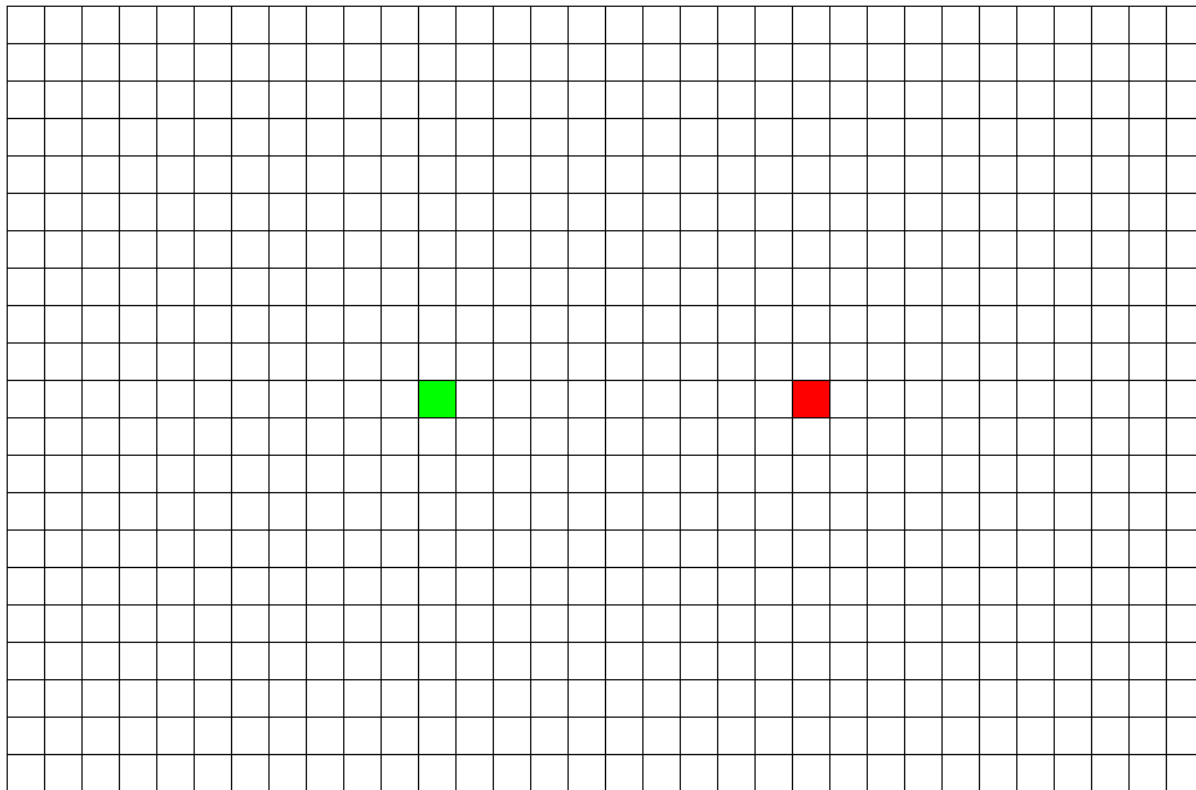
学 | 竞 | 赛 |
Southwest University



更优的搜索方式



西南大学附属中学
High School Affiliated to Southwest University

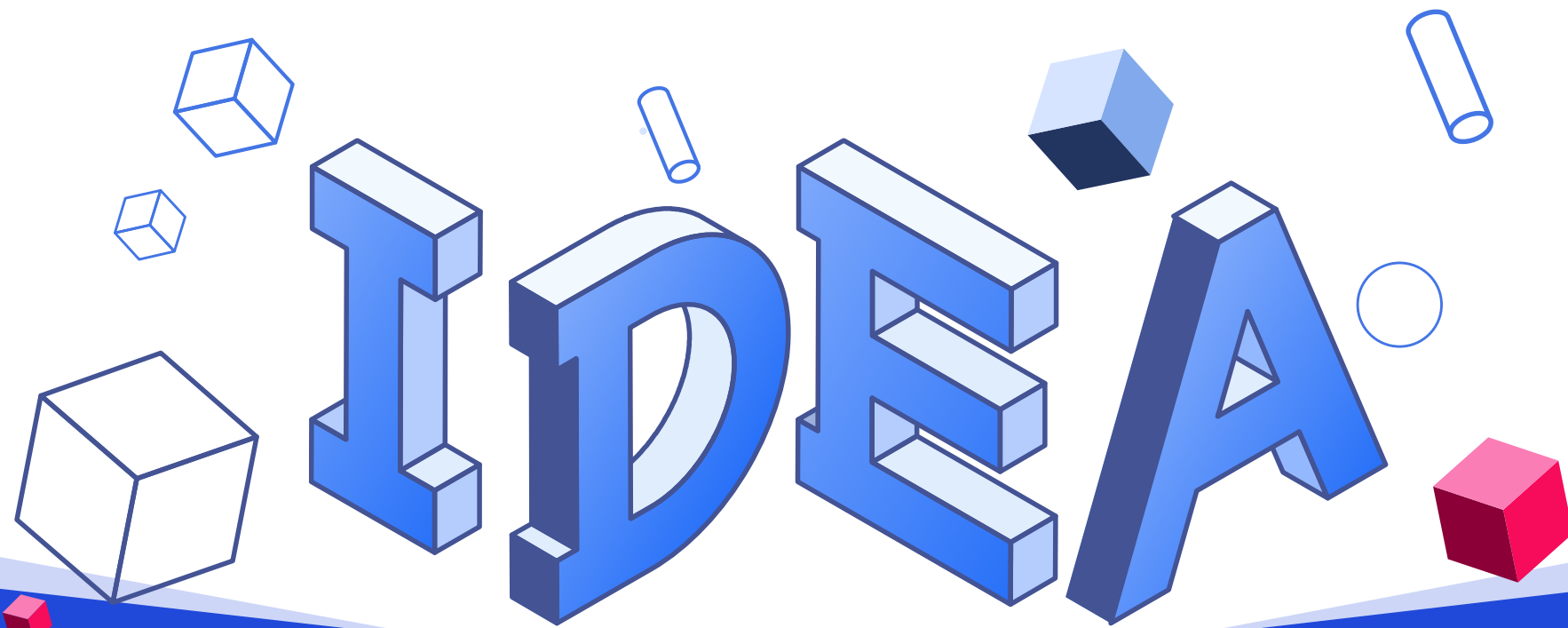


与DFS一条路走到黑不同，

类似于这样一层一层的往外搜索的方式

称为“广度优先搜索” (BFS)

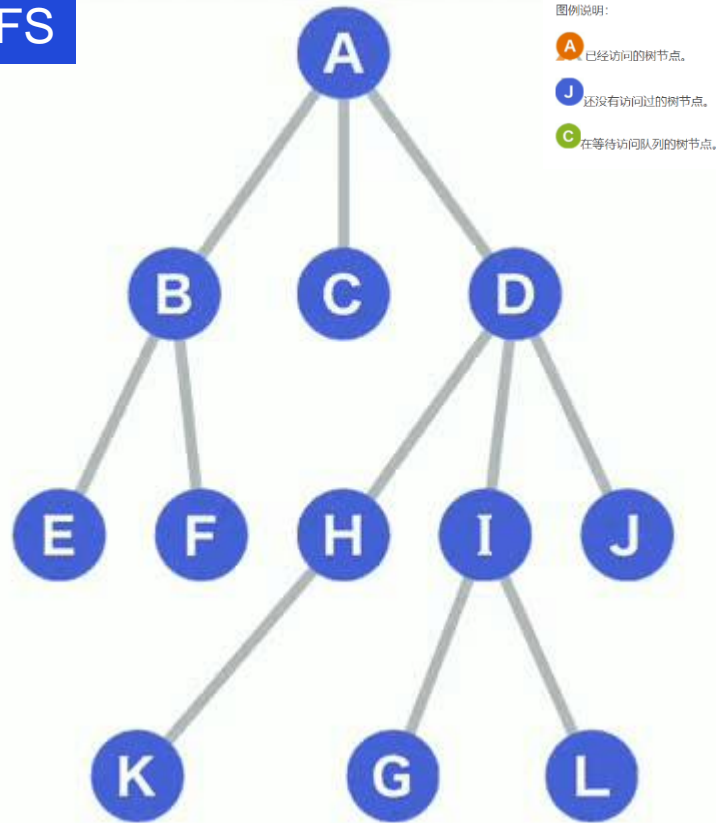
BFS能保证第一次搜到终点的路径就是最短的



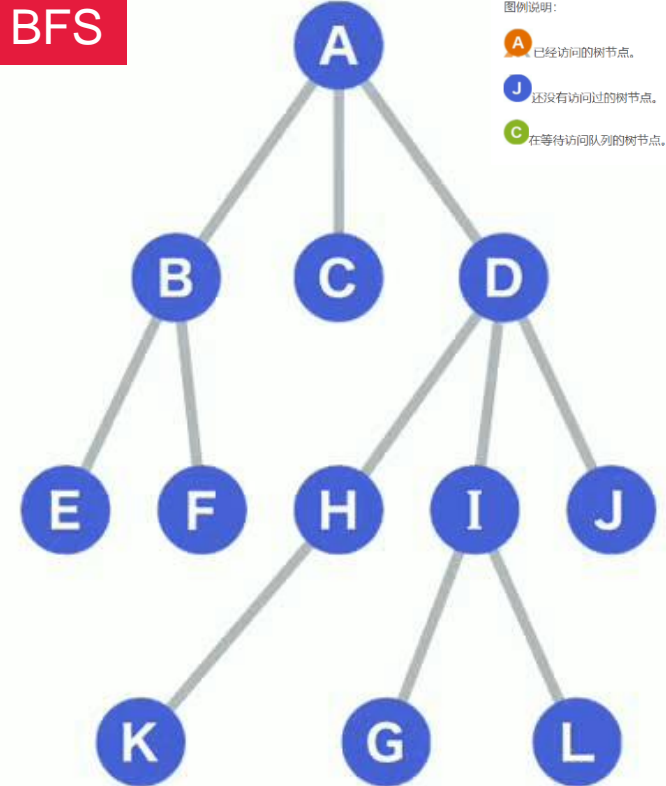
信息学 广度优先搜索 (Broad First Search)

西南大学附属中学校

信息奥赛教练组



结点访问的特点：搜索结点先访问的后退出 基于递归栈实现



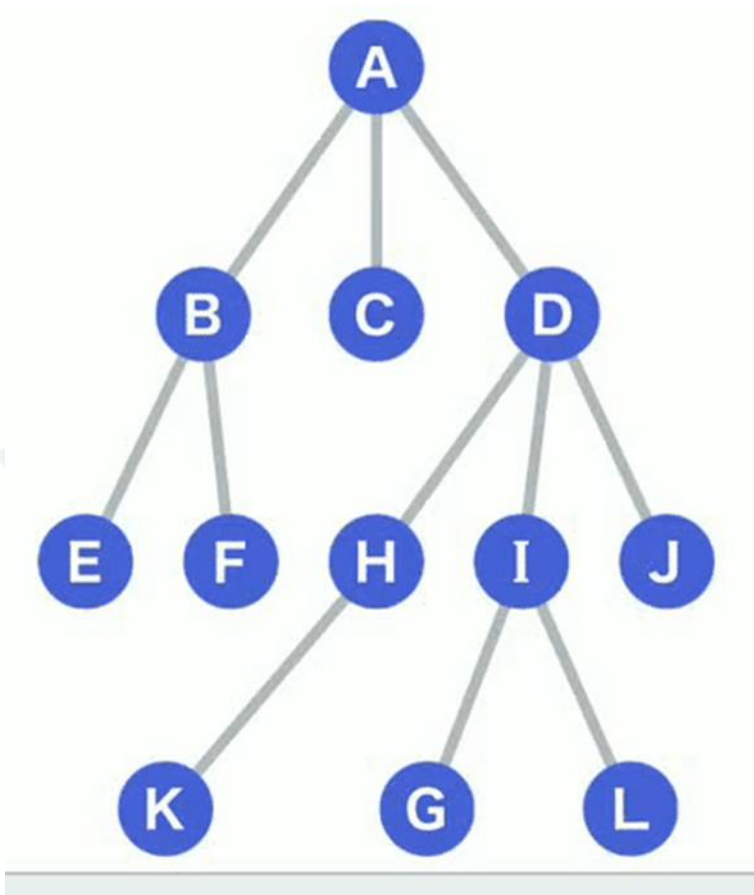
结点访问的特点：搜索结点先访问的先退出
基于队列实现



BFS模板



西南大学附属中学
High School Affiliated to Southwest University



算法流程

存储起始结点信息(把起点信息放入队列)

```
while(队列不为空){
```

```
    取出队首元素u;
```

```
    标记为出队;
```

```
    将队首元素出队;
```

```
    for(遍历u的下一层未曾入队的结点){
```

```
        if(下层结点满足条件) {
```

```
            保存结点信息, 结点入队
```

```
            结点标记为已入队
```

```
        }
```

```
    }
```

```
}
```

入队过程:

A

B

C

D

E

F

H

I

J

K

G

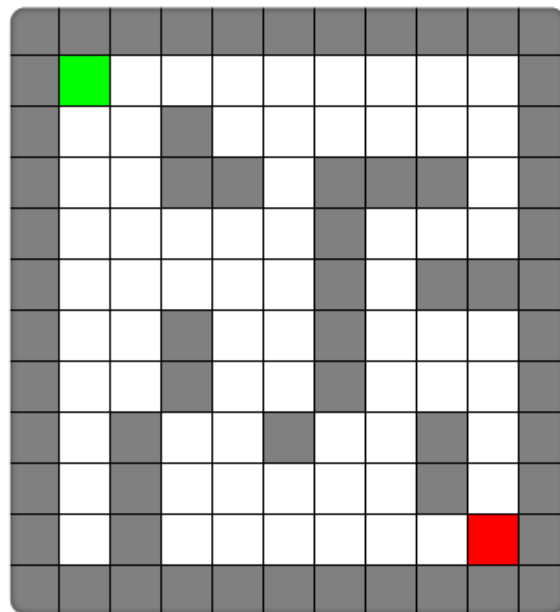
L



1. 迷宫最短路问题

求解起点到终点的最短距离

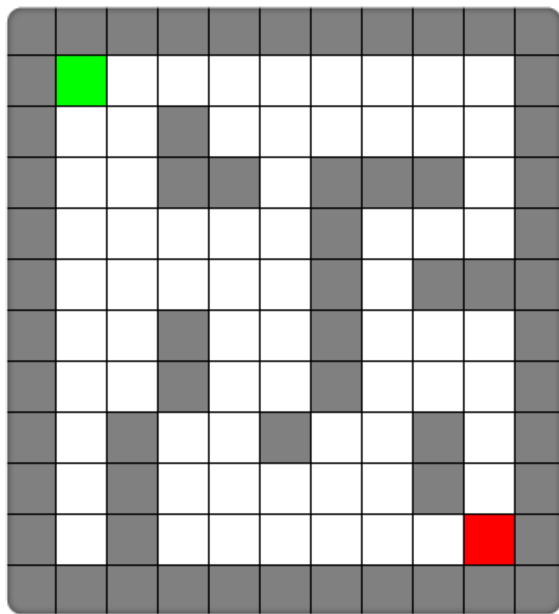
DFS+剪枝有时也能做





2.同时移动问题

起始点火山喷发了，主角开始逃离
在 t 时间后熔岩会到达终点，
问主角能否在 t 时间内抵达？

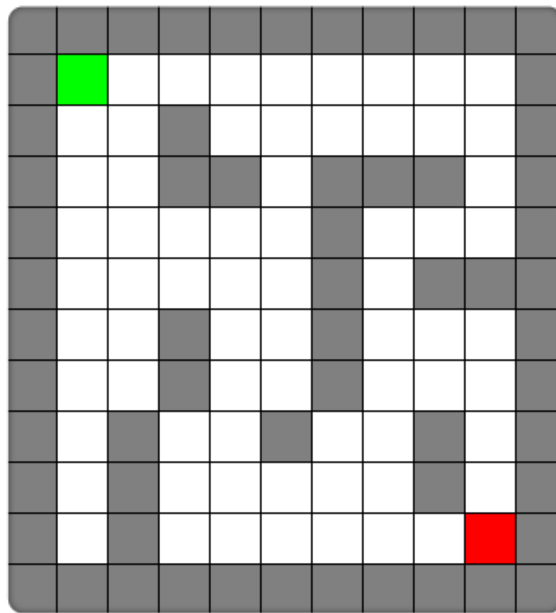


信息学竞赛
Southwest University



3. 路径还原问题

不光问最短路径的长度，还要输出最短路径的信息



言 | 息 | 学 | 竞 | 赛 |

High School Affiliated to Southwest University



4.倒水次数问题

参见《非常可乐》题面，
求最少的倒水次数

关键信息：最少次数(步数)，最短路径

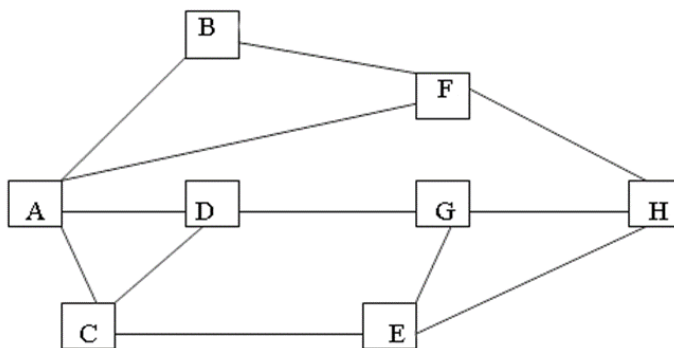


例：最短路径问题(1)



西南大学附属中学
High School Affiliated to Southwest University

下图表示的是从城市A到城市H的交通图。从图中可以看出，从城市A到城市H要经过若干个城市。现给给出一个城市的路线图，并给出起点与终点，请给出最短的距离。



【输入格式】

第一行三个数，分别是起点城市，目标城市，线路条数n。
接下来n行，每行两个数字，表示这两个城市之间有线路。

注意：输入数据中1表示城市A，2表示城市B.....（你应该知道数据范围了吧？）

【输出格式】

输出路线一行，用->表示箭头，具体格式见样例。

样例输入

1 8 12

1 2

1 3

1 4

1 6

2 6

3 4

3 5

4 7

5 7

5 8

6 8

7 8

样例输出

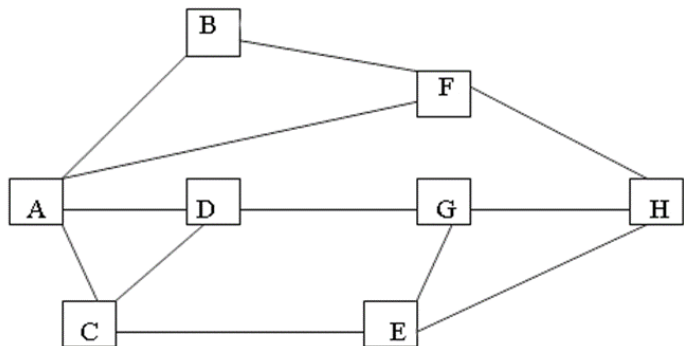
3



分析



西南大学附属中学
High School Affiliated to Southwest University



存储起始结点信息(把起点信息放入队列)

```
while(队列不为空){
```

```
    取出队首元素u;
```

```
    标记为出队;
```

```
    将队首元素出队;
```

```
    for(遍历u的下一层未曾入队的结点){
```

```
        if(下层结点满足条件) {
```

```
            结点入队
```

```
            结点标记为已入队
```

```
        }
```

```
    }
```

```
}
```

结点包含的信息有:

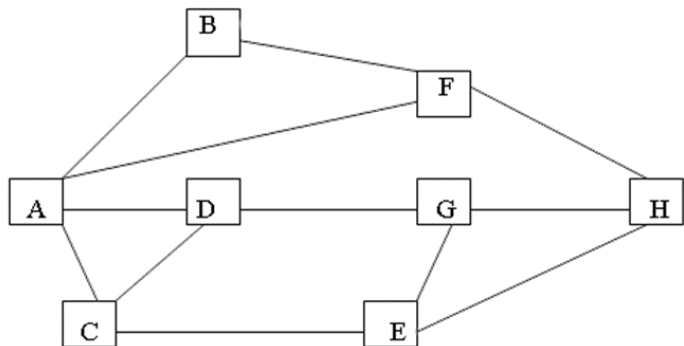
起点到结点i的距离dis[i]

结点i是否在队列中vis[i]

西 | 大 | 附 | 中 | 信 | 息 | 学 | 竞 | 赛 |
High School Affiliated to Southwest University



分析



存储起始结点信息(把起点信息放入队列)

while(队列不为空){

取出队首元素u;

标记为出队;

将队首元素出队;

for(遍历u的下一层未曾入队的结点){

if(下层结点满足条件) {

保存结点信息, 结点入队

结点标记为已入队

}

}

}

```
int main(){
    int st,ed,n;
    scanf("%d %d %d",&st,&ed,&n); //输入数据
    for(int i=0;i<n;i++){
        int x,y;
        scanf("%d%d",&x,&y);
        city[x][y] = city[y][x] = 1; //存储连接关系
    }

    int head=tail=1;
    q[tail]=st; //存储起始点信息
    dis[st]=0;
    vis[st]=1;
    tail++;
    while(head != tail){
        int u=q[head]; //取出队首u结点信息
        int step=dis[head];
        vis[u]=0; //表示队首u已出队
        head++;
        for(int i=1; i<=26; i++){
            if(city[u][i]==1&&vis[i]==0){ //联通且未入队
                q[tail]=i; dis[tail]=step+1;
                vis[tail]=1;
                tail++;
                if(i==ed) {cout<<step;exit(0);}
            }
        }
    }
}
```

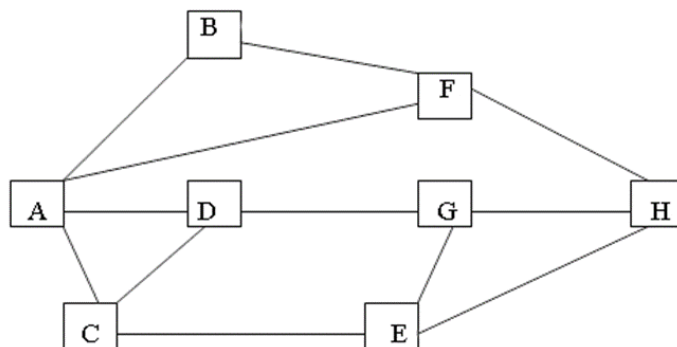


例：最短路径问题(2)



西南大学附属中学
High School Affiliated to Southwest University

下图表示的是从城市A到城市H的交通图。从图中可以看出，从城市A到城市H要经过若干个城市。现给给出一个城市的路线图，并给出起点与终点，请给出最短的路线。



【输入格式】

第一行三个数，分别是起点城市，目标城市，线路条数n。
接下来n行，每行两个数字，表示这两个城市之间有线路。

注意：输入数据中1表示城市A，2表示城市B.....（你应该知道数据范围了吧？）

【输出格式】

输出路线一行，用->表示箭头，具体格式见样例。

样例输入

1 8 12

1 2

1 3

1 4

1 6

2 6

3 4

3 5

4 7

5 7

5 8

6 8

7 8

样例输出

A->F->H

再定义一个pre[], 记录它是由哪个点转移过来的

最后程序运行的结果

i下标	1	2	3	4	5	6	7	8
pre数组	2	1	1	1	3	1	4	6

如何读?

8 的前一个点是 6

6 的前一个点是 1

Q: 知道终点如何逆推出路径

递归输出

```
void print(int ed){  
    if(ed!=1){  
        print(pre[ed]);  
        cout<<"->"<<(char)(ed + 'A'-1);  
    }  
}
```




核心代码



西南大学附属中学
High School Affiliated to Southwest University

```
int main(){
    int st,ed,n;
    scanf("%d%d%d",&st,&ed,&n);

    for(int i=0;i<n;i++){
        int x,y;
        scanf("%d%d",&x,&y);
        city[x][y] = city[y][x] = 1;
    }

    int head=tail=1;
    q[tail]=st;
    vis[st]=1;
    pre[st]=0;
    tail++;
    while(head != tail){
        int u=q[head];
        vis[u]=0;
        head++;
        for(int i=1; i<=26 ;i++){
            if(city[u][i]==1&&pre[i]==0){
                q[tail]=i;
                pre[tail]=u;
                tail++;
            }
        }
    }
    printf("%c",st+'A'-1); //单独输出一下起点A
    print(ed);
}
```

```
void print(int ed){
    if(ed!=1){
        print(pre[ed]);
        cout<<"->"<<(char)(ed + 'A'-1);
    }
}
```

也可以转化为非递归版本的输出

西 | 大 | 附 | 中 | 信 | 息 | 学 | 竞 | 赛 |
High School Affiliated to Southwest University



例：奇怪的电梯



西南大学附属中学
High School Affiliated to Southwest University

大楼的每一层楼都可以停电梯，而且第 i 层楼 ($1 \leq i \leq N$) 上有一个数字 K_i ($0 \leq K_i \leq N$)。

电梯只有四个按钮：开，关，上，下。上下的层数等于当前楼层上的那个数字。

当然，如果不能满足要求，相应的按钮就会失灵。

例如：3 3 1 2 5代表了 K_i ($K_1=3, K_2=3, \dots$)，从一楼开始。在一楼，按“上”可以到4楼，按“下”是不起作用的，因为没有-2楼。那么，从A楼到B楼至少要按几次按钮呢？

输入

输入共有二行，第一行为三个用空格隔开的正整数，表示
 N, A, B ($1 \leq N \leq 200, 1 \leq A, B \leq N$)，第二行为 N 个用空格隔开的正整数，
表示 K_i 。

输出

输出文件仅一行，即最少按键次数,若无法到达，则输出-1

样例输入

5 1 5

3 3 1 2 5

样例输出

3

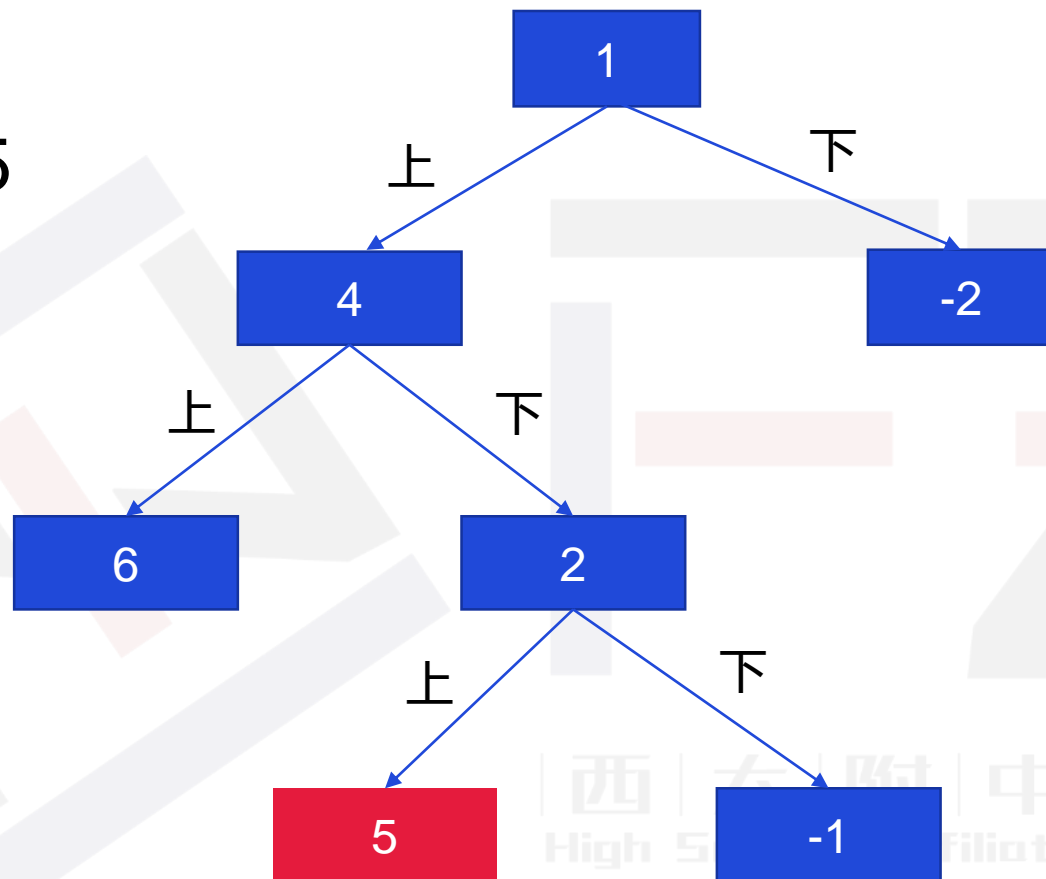


分析



西南大学附属中学
High School Affiliated to Southwest University

5 1 5
3 3 1 2 5



每个节点只有两种状态：上，下



参考代码



西南大学附属中学
High School Affiliated to Southwest University

```
int n,m,a,b;
int q[250],vis[250];
queue<int> p;
int main()
{
    scanf("%d %d %d",&n,&a,&b);
    for(int i=1;i<=n;i++) scanf("%d",&q[i]);
    vis[a]=1; //把标记和步数统计合二为一
    p.push(a);
    while(!p.empty()){
        int id=p.front();
        int down= id -q[id],up= id +q[id];
        if(down >=1&&!vis[down]){ //电梯向下
            p.push(down);
            vis[down]=vis[id]+1;
        }
        if(up <=n&&!vis[up]){ //电梯向上
            p.push(up);
            vis[up]=vis[id]+1;
        }
        if(down==b || up==b)break; //到达b
        p.pop();
    }
    printf("%d\n",vis[b]-1);
    return 0;
}
```



西 | 大 | 附 | 中 | 信 | 息 | 学 | 竞 | 赛 |
High School Affiliated to Southwest University



可以解决的问题



西南大学附属中学
High School Affiliated to Southwest University

1. 迷宫最短路问题
2. 同时移动问题
3. 倒水问题
4. 路径还原问题

各种路径搜索算法讲解：

https://zhuanlan.zhihu.com/p/346666812?utm_medium=social&utm_oi=1096915858005323776

Thanks

For Your Watching

