程序设计实习

郭炜　微博 http://weibo.com/guoweiofpku
http://blog.sina.com.cn/u/3266490431
刘家瑛 微博 http://weibo.com/pkuliujiaying

广度优先搜索

入门：抓住那头牛

抓住那头牛(POJ3278)

农夫知道一头牛的位置，想要抓住它。农夫和牛都位于数轴上，农夫起始位于点N(0<=N<=100000)，牛位于点K(0<=K<=100000)。农夫有两种移动方式：

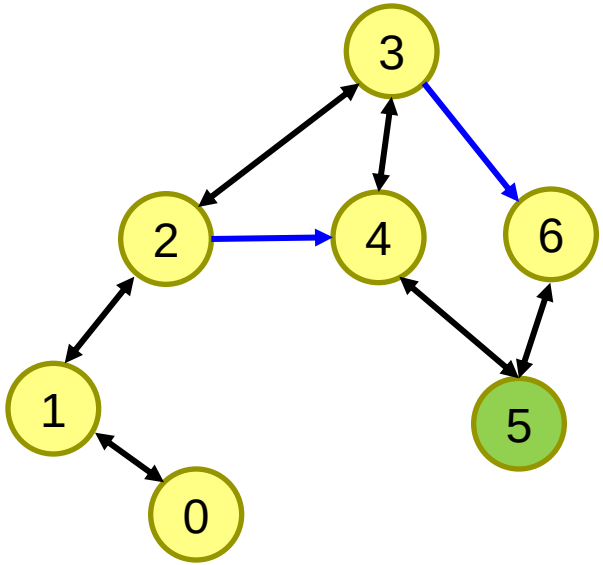
1、从X移动到X-1或X+1，每次移动花费一分钟
2、从X移动到2*X，每次移动花费一分钟

假设牛没有意识到农夫的行动，站在原地不动。农夫最少要花多少时间才能抓住牛？

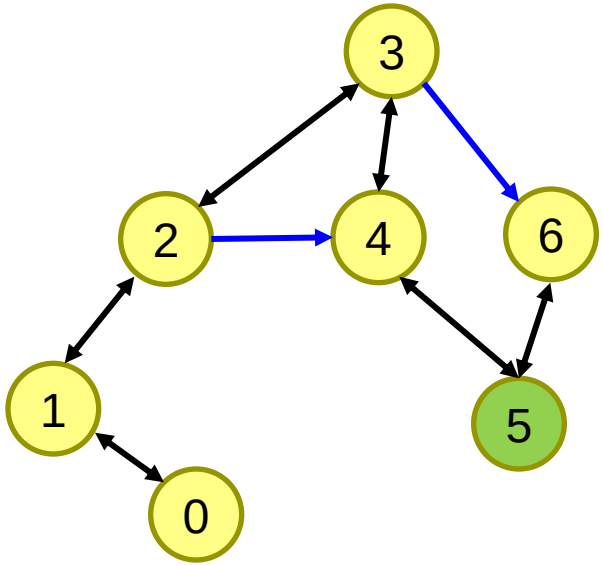
假设农夫起始位于点3，牛位于5

N=3, K=5，最右边是6。

如何搜索到一条走到5的路径？

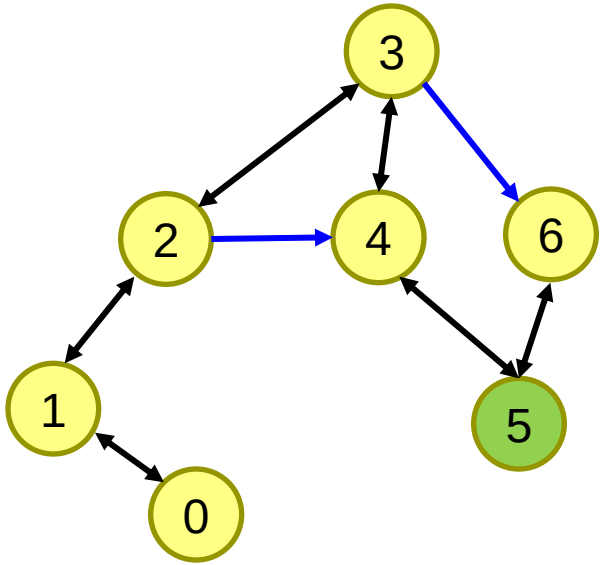


假设农夫起始位于点3，牛位于5
N=3, K=5，最右边是6。
如何搜索到一条走到5的路径？



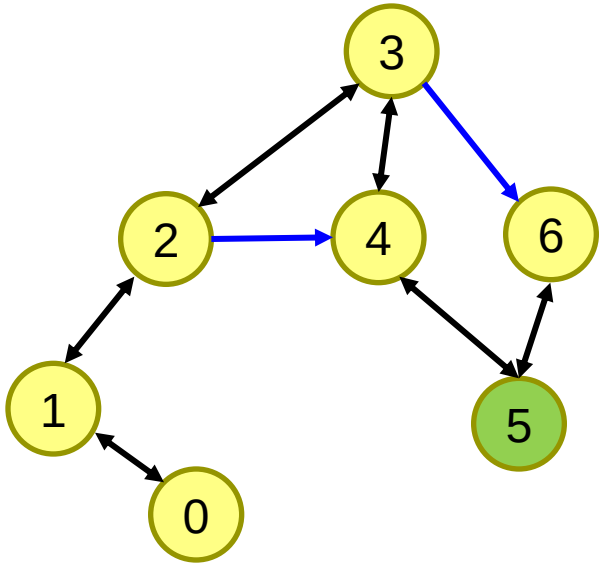
策略1）深度优先搜索：从起点出发，随机挑一个方向，能往前走就往前走(扩展)，走不动了则回溯。不能走已经走过的点(要判重)。

假设农夫起始位于点3，牛位于5
N=3, K=5，最右边是6。
如何搜索到一条走到5的路径？



运气好的话：
3->4->5
或
3->6->5
问题解决！

# 假设农夫起始位于点3，牛位于5
N=3, K=5，最右边是6。
如何搜索到一条走到5的路径？

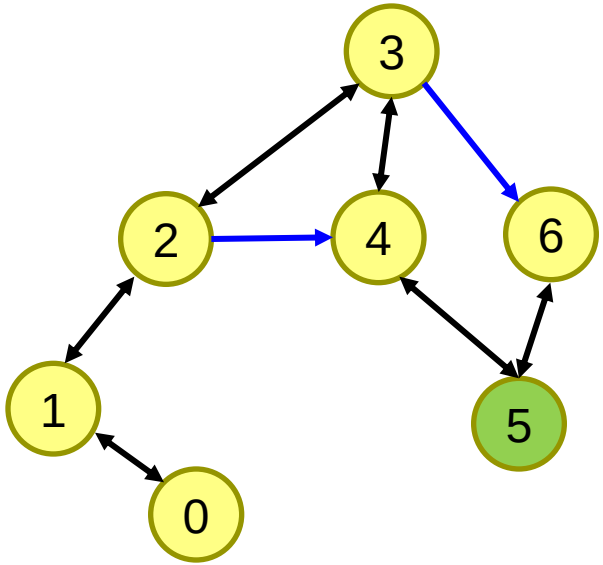


运气不太好的话：
3->2->4->5
运气最坏的话：
3->2->1->0->4->5

要想求最优(短）解，则要遍历所有走法。可以用各种手段优化，比如，若已经找到路径长度为n的解，则所有长度大于n的走法就不必尝试。

运算过程中需要存储路径上的节点，数量较少。用栈存节点。

假设农夫起始位于点3，牛位于5
N=3, K=5，最右边是6。
如何搜索到一条走到5的路径？

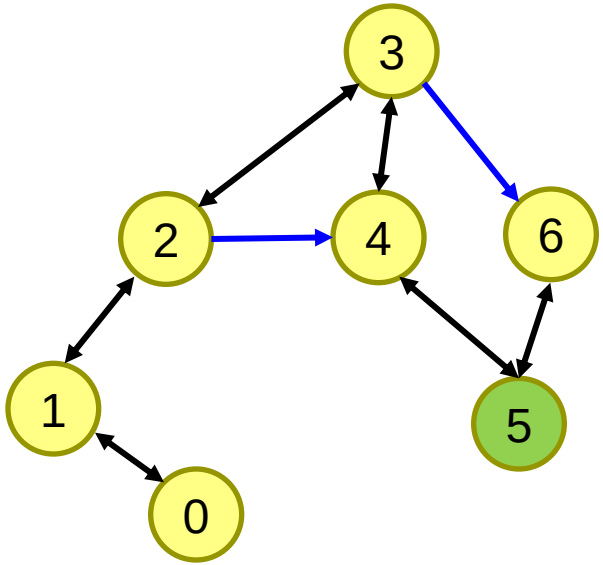


策略2）广度优先搜索：

给节点分层。起点是第0层。从起点最少需n步就能到达的点属于第n层。

第1层：2, 4, 6
第2层：1, 5
第3层：0

假设农夫起始位于点3，牛位于5
N=3, K=5，最右边是6。
如何搜索到一条走到5的路径？

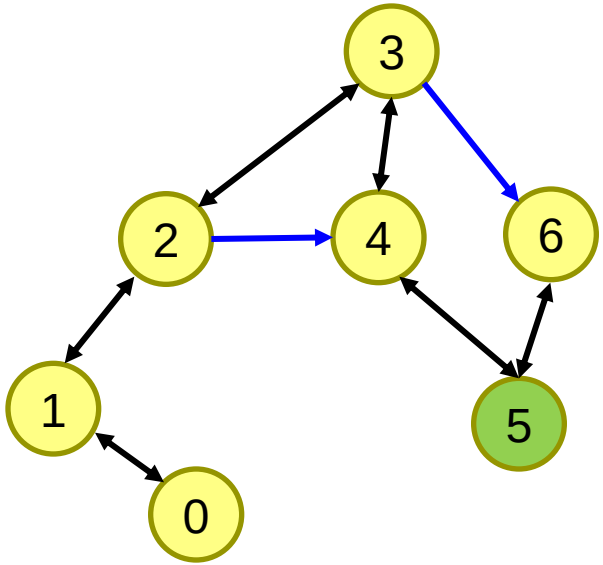


策略2）广度优先搜索：

给节点分层。起点是第0层。从起点最少需n步就能到达的点属于第n层。

依层次顺序，从小到大扩展节点。把层次低的点全部扩展出来后，才会扩展层次高的点。

假设农夫起始位于点3，牛位于5
N=3, K=5，最右边是6。
如何搜索到一条走到5的路径？



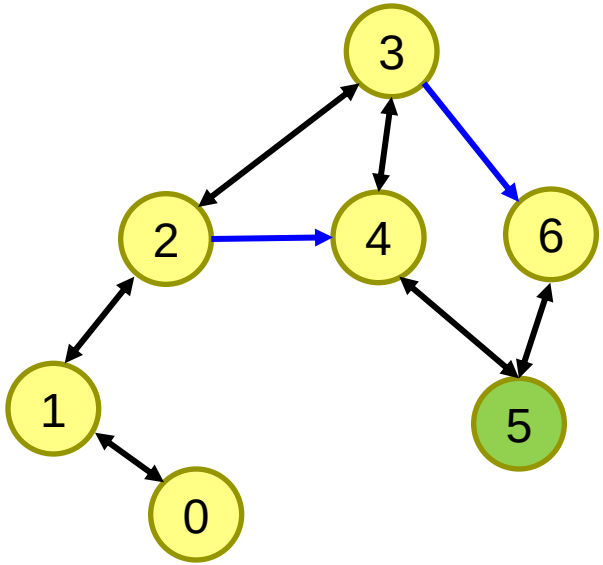
策略2）广度优先搜索：

搜索过程（节点扩展过程）：

3
2 4 6
1 5

问题解决。
扩展时，不能扩展出已经走过的节点(要判重) 。

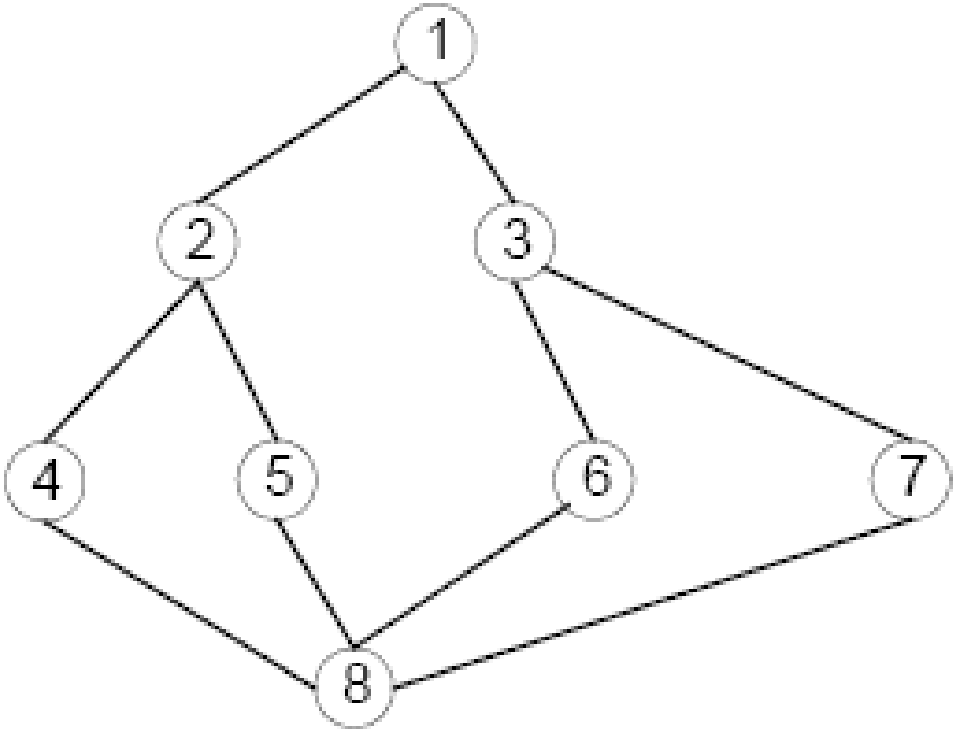
假设农夫起始位于点3，牛位于5
N=3, K=5，最右边是6。
如何搜索到一条走到5的路径？



策略2）广度优先搜索：

可确保找到最优解，但是因扩展出来的节点较多，且多数节点都需要保存，因此需要的存储空间较大。
用队列存节点。

深搜 vs. 广搜



若要遍历所有节点：

- 深搜

1-2-4-8-5-6-3-7

- 广搜

1-2-3-4-5-6-7-8

广搜算法

□广度优先搜索算法如下：（用QUEUE）

(1) 把初始节点S0放入Open表中；

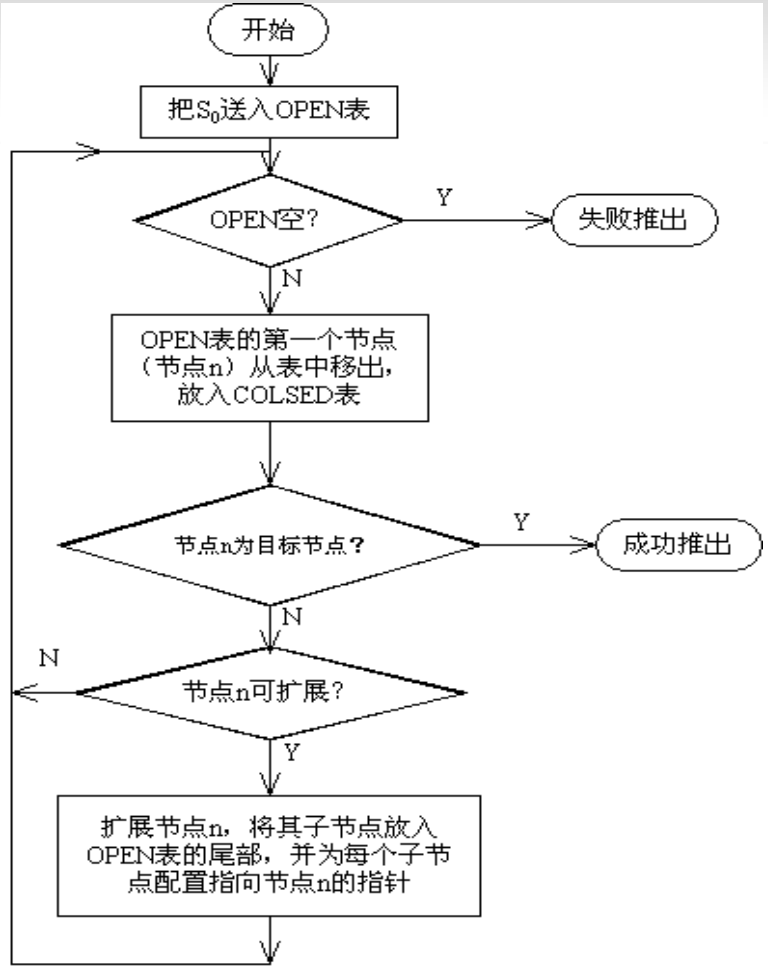
(2) 如果Open表为空，则问题无解，失败退出；

(3) 把Open表的第一个节点取出放入Closed表，并记该节点为n；

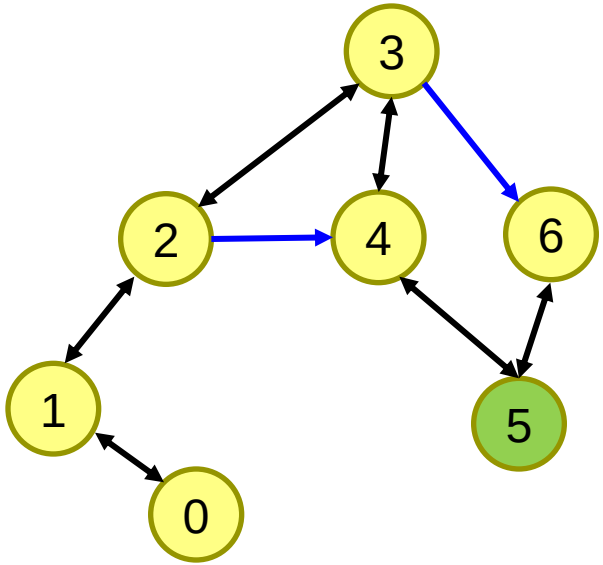
(4) 考察节点n是否为目标节点。若是，则得到问题的解，成功退出；

(5) 若节点n不可扩展，则转第(2)步；

(6) 扩展节点n，将其不在Closed表和Open表中的子节点(判重）放入Open表的尾部，并为每一个子节点设置指向父节点的指针(或记录节点的层次），然后转第(2)步。



假设农夫起始位于点3，牛位于5
N=3, K=5，最右边是6。
如何搜索到一条走到5的路径？



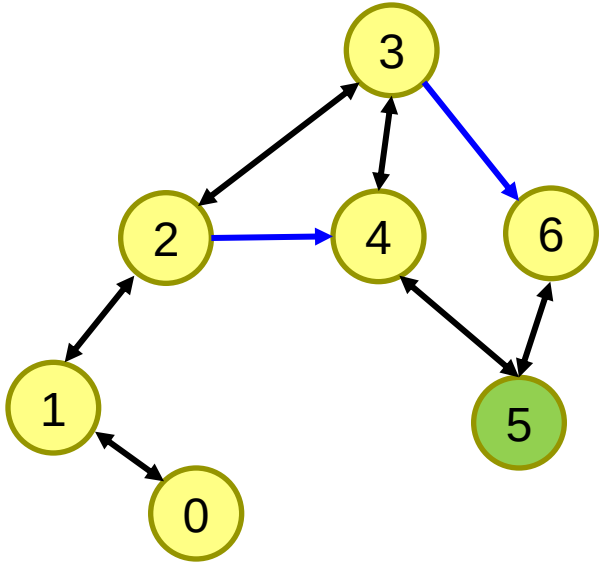
广度优先搜索队列变化过程：

Closed

3

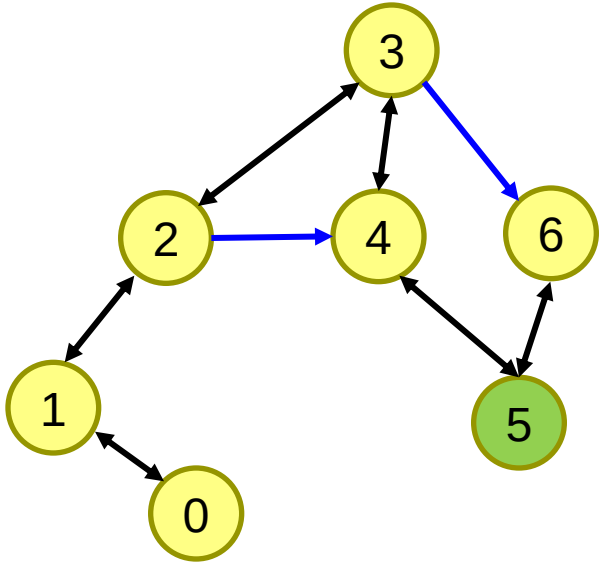
Open

假设农夫起始位于点3，牛位于5
N=3, K=5，最右边是6。
如何搜索到一条走到5的路径？

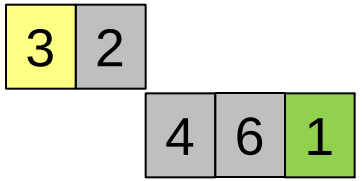


广度优先搜索队列变化过程：

假设农夫起始位于点3，牛位于5
N=3, K=5，最右边是6。
如何搜索到一条走到5的路径？



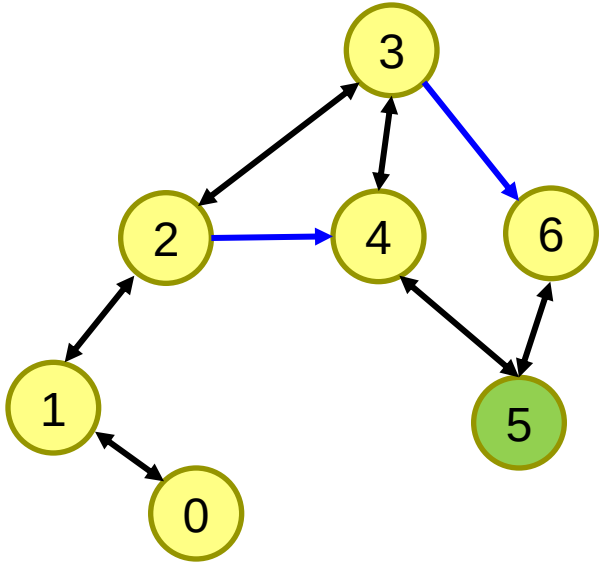
广度优先搜索队列变化过程：



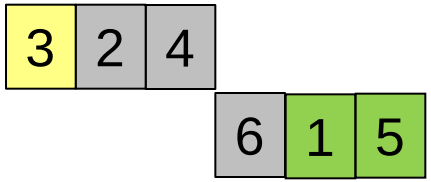
Closed

Open

假设农夫起始位于点3，牛位于5
N=3, K=5，最右边是6。
如何搜索到一条走到5的路径？



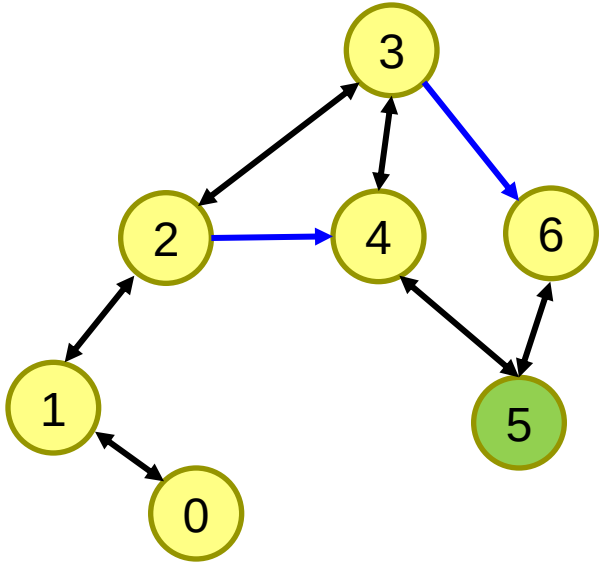
广度优先搜索队列变化过程：



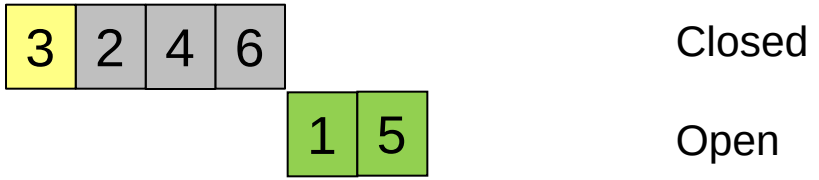
Closed

Open

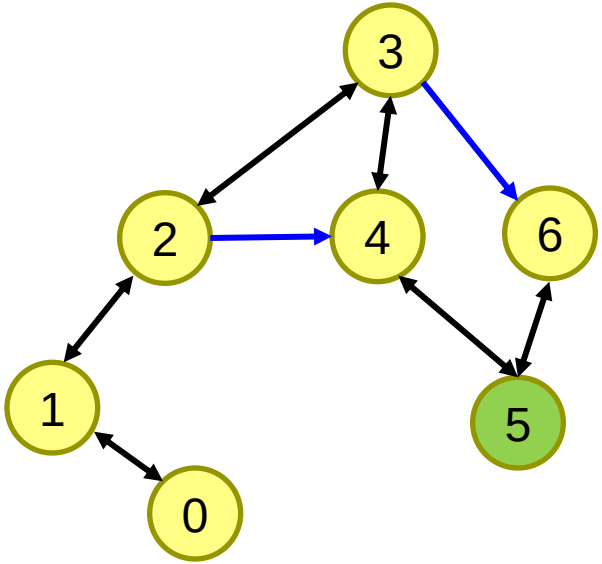
假设农夫起始位于点3，牛位于5
N=3, K=5，最右边是6。
如何搜索到一条走到5的路径？



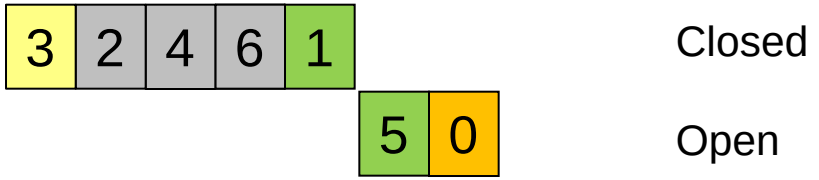
广度优先搜索队列变化过程：



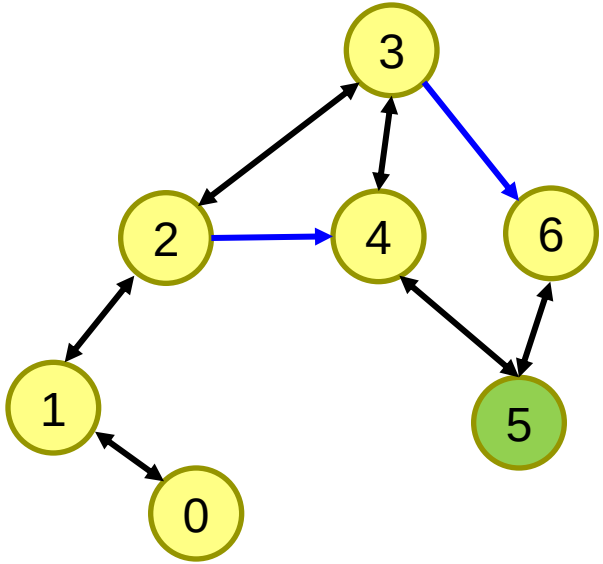
假设农夫起始位于点3，牛位于5
N=3, K=5，最右边是6。
如何搜索到一条走到5的路径？



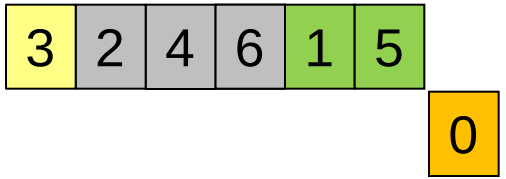
广度优先搜索队列变化过程：



假设农夫起始位于点3，牛位于5
N=3, K=5，最右边是6。
如何搜索到一条走到5的路径？



广度优先搜索队列变化过程：



目标节点5出队列，问题解决！

//poj3278 Catch That Cow

```
#include <iostream>
#include <cstring>
#include <queue>
using namespace std;
int N,K;
const int MAXN = 100000;
int visited[MAXN+10];  //判重标记,visited[i] = true表示i已经扩展过
struct Step{
        int x;  //位置
        int steps;  //到达x所需的步数
        Step(int xx,int s):x(xx),steps(s) { }
};
queue<Step> q;  //队列,即Open表
int main()   {
        cin >> N >> K;
        memset(visited,0,sizeof(visited));
        q.push(Step(N,0));
        visited[N] = 1;
```

```
while(!q.empty()) {
        Step s = q.front();
        if( s.x == K ) { //找到目标
                cout << s.steps <<endl;
                return 0;
        }
        else {
                if( s.x - 1 >= 0 && !visited[s.x-1] ) {
                        q.push(Step(s.x-1,s.steps+1));
                        visited[s.x-1] = 1;
                }
                if( s.x + 1 <= MAXN && !visited[s.x+1] ) {
                        q.push(Step(s.x+1,s.steps+1));
                        visited[s.x+1] = 1;
                }
```

```
                    if( s.x * 2 <= MAXN &&!visited[s.x*2]  ) {
                        q.push(Step(s.x*2,s.steps+1));
                        visited[s.x*2]  = 1;
                    }
                    q.pop();
                }
            }
            return 0;
}
```