

1253: 抓住那头牛

题目描述

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 ($0 \leq N \leq 100000$)，牛位于点 K ($0 \leq K \leq 100000$)。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 $2 \times X$ ，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？

输入格式

两个整数， N 和 K 。

输出格式

一个整数，农夫抓到牛所要花费的最小分钟数。

输入样例

5 17

输出样例

4

解析

一维地图，但是存在三种移动方式： $x+1$ 、 $x-1$ 、 $x * 2$ 。求最短路径，很明显需要使用广搜。

编码

```
#include <bits/stdc++.h>

using namespace std;

int const MaxNum = 100001;

//节点结构体
struct Node {
    int x;    //坐标
    int step; //移动步数
    //当前节点的基本信息
```

```

Node(int nx, int stepNum) {
    x = nx;
    step = stepNum;
}

Node() = default;
};

int N, K;    //农夫和牛的位置
bool Vis[MaxNum];    //判断是否走过
int gapX[3] = {-1, 1, 2};    //三种移动方式
Node NodeQueue[MaxNum];    //中间阶段的数据，要尽可能的大

//检测是否可以通过
bool CanPass(int newX) {
    //保证没有越界
    if (newX >= 0 && newX <= MaxNum) {
        //没有被访问过
        if (!Vis[newX]) {
            return true;
        }
    }
    return false;
}

//广度搜索
void Bfs() {
    Node start = Node(N, 0);
    Node end = Node(K, 1);
    //初始化
    memset(Vis, false, sizeof(Vis));
    memset(NodeQueue, 0, sizeof(NodeQueue));
    //设置头尾
    int head = 0;
    int tail = 0;
    //将首点加入队列
    NodeQueue[head] = start;
    //设置该节点已经被访问
    Vis[start.x] = true;
    //标记当前节点已经走过
    while (head <= tail) {
        //取出一个节点
        Node room = NodeQueue[head];
        head++;
        //三种移动方式
        for (int k = 0; k < 3; ++k) {
            int newX = room.x + gapX[k];

```

```

        if (k == 2) {
            newX = room.x * 2;
        }
        //找到目标;
        if (newX == end.x) {
            //输出一共走的步数记录最后一个点
            cout << room.step + 1 << endl;
            return;
        }
        //下个节点可以被访问
        if (CanPass(newX)) {
            //设置该节点已经被访问
            Vis[newX] = true;
            tail++;
            //将数据加入新的队列
            NodeQueue[tail] = Node(newX, room.step + 1);
        }
    }
}
//没有找到路线
cout << -1 << endl;
}

//读取输入的数据
void ReadInfo() {
    cin >> N >> K;
}

int main() {
    ReadInfo();
    Bfs();
    return 0;
}

```

逻辑航线培优教育，信息学奥赛培训专家。

扫码添加作者获取更多内容。

