

1262: 挖地雷

题目描述

在一个地图上有 n 个地窖 ($n \leq 200$)，每个地窖中埋有一定数量的地雷。同时，给出地窖之间的连接路径，并规定路径都是单向的,且保证都是小序号地窖指向大序号地窖，也不存在可以从一个地窖出发经过若干地窖后又回到原来地窖的路径。某人可以从任意一处开始挖地雷，然后沿着指出的连接往下挖（仅能选择一条路径），当无连接时挖地雷工作结束。设计一个挖地雷的方案，使他能挖到最多的地雷。

输入

第一行：地窖的个数；

第二行：为依次每个地窖地雷的个数；

下面若干行：

$x_i y_i$ //表示从 x_i 可到 y_i ， $x_i < y_i$ 。

最后一行为"00"表示结束。

输出

$k_1-k_2-\dots-k_v$ //挖地雷的顺序

挖到最多的雷。

输入样例

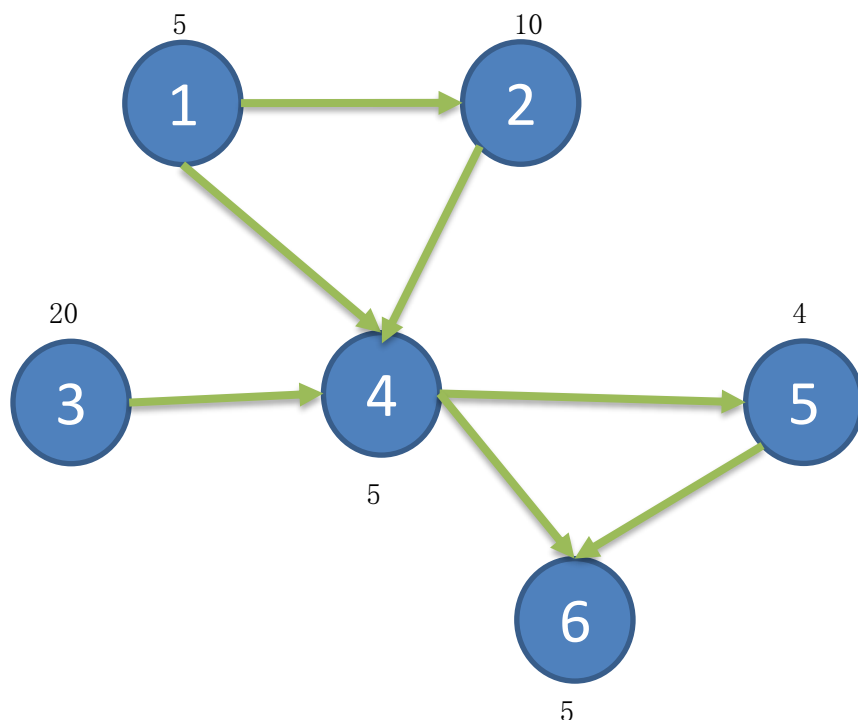
```
6
5 10 20 5 4 5
1 2
1 4
2 4
3 4
4 5
4 6
5 6
0 0
```

输出样例

```
3-4-5-6
34
```

解析

根据题意，我们将图像绘制出来，如下所示：



初看题目，我们很容易想到，最简单的办法就是找出全部的路径，然后将其上面的地雷数进行累加，但是，很明显这样会超限，因为计算量太大。因此，正确的方法还是动态规划。

首先，我们定义 $dp[i]$ ，表示以 i 为结尾的地雷数。定义 num 数组表示每个地雷的地雷数。

则有， $dp[6] = \max(dp[5] + num[6], dp[4] + num[6])$ 。

继续深入推导，可以发现，想要知道 $dp[6]$ 必须知道 $dp[5]$ ，想要知道 $dp[5]$ 则必须知道 $dp[4]$...实际上我们是知道 $dp[1]$ 的，它的数值就是 $num[1]$ 。

根据以上原理，我们可以解决本问题。

编码

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

#define INF 0x3f3f3f3f
#define MAXN 210
using namespace std;
int n, a[MAXN][MAXN], num[MAXN], dp[MAXN], p[MAXN];

/*a数组用于标识通路
  num数组用于标识地雷数
  dp数组用于标识后继的最大地雷数
  p数组用于存取最优路线
*/
```

```

void print(int k) {
    if (k == 0) return;
    print(p[k]);
    if (p[k] == 0) cout << k;    //如果是第一个点直接输出k
    else cout << "-" << k;      //否则输出-k
}

int main() {
    memset(a, 0, sizeof(a));
    memset(dp, 0, sizeof(dp));
    memset(p, 0, sizeof(p));
    cin >> n;
    for (int i = 1; i <= n; i++) {
        cin >> num[i];          //输入地雷数
        dp[i] = num[i];          //设置dp数组初值
    }
    int x, y;
    while (cin >> x >> y, x, y) a[x][y] = 1;    //标识通路
    //将ans设置为一个足够小的数，k用于记录最后挖的地窖
    int ans = -INF, k = 0;
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= n; j++) {
            //通路且dp[i]<dp[j]+num[i]
            if (a[j][i] == 1 && dp[i] < dp[j] + num[i]) {
                dp[i] = dp[j] + num[i];
                p[i] = j;          //记录最优路线
            }
        }
        if (dp[i] > ans) {          //获取最大地雷数
            ans = dp[i];           //ans即为最大地雷数
            k = i;                 //k用来记录最后一个路线
        }
    }
    print(k);    //输出最优路线
    cout << endl << ans << endl;
    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。

