算法设计与分析 C4-B

21377206 阮阳栋

题目描述

给定 $n \times m$ 个格子，第 i 行第 j 列（$1 \leq i \leq n, 1 \leq j \leq m$）的格子记为$(i, j)$，有权值 a_{ij}。

初始时你在$(1, 1)$。当你在(i, j)时，你可以任意选择目标格子(x, y)（$1 \leq x \leq n, 1 \leq y \leq m$），并支付$|a_{ij} - a_{xy}|$的费用到达$(x, y)$。

试求你从$(1, 1)$到达(n, m)的最小费用。

题目分析

题目是求最小费用，可以转化为最短路问题。这$n \times m$个点两两有无向边，也有权值，现在要求起点到终点的最短路。由于是无向图，可以考虑Bellman-Ford算法或者Dijskra算法。很显然这两种算法都不可以解决这个问题，因为这个题目的无向边个数为$O(n^2m^2)$，实在是太多了。

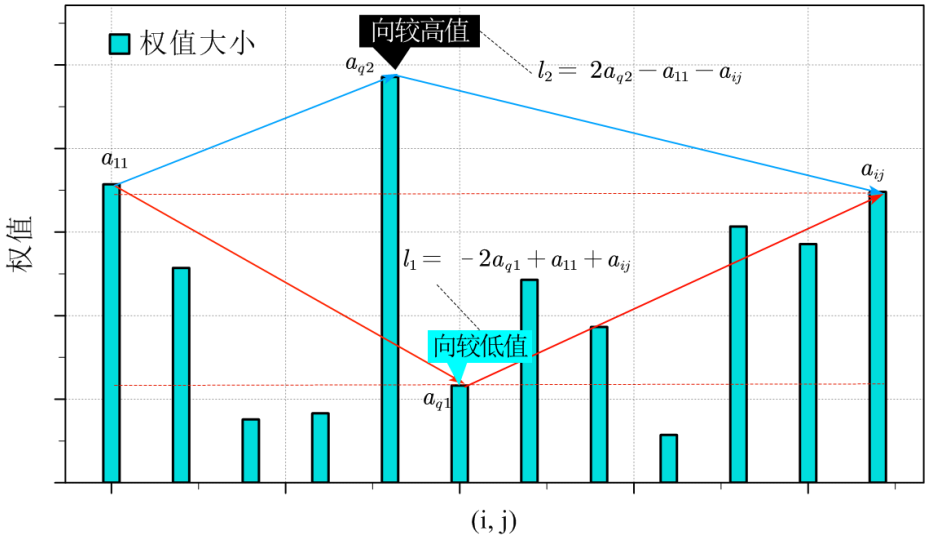
换一个思路，如果把这个问题当成动态规划也可以。给出状态转移方程：

$$dp(i,j) = \begin{cases} 0 & i = j = 1 \\ \min\{dp(k,q) + |a_{ij} - a_{kq}|\} & 1 \leq k \leq i, 1 \leq q \leq j, k \neq i \vee q \neq \end{cases}$$

```
for(int i=1;i<=n;i++)for(int j=1;j<=m;j++){
    for(int k=1;k<=i;k++)for(int q=1;q<=j;q++){
        dp[i][j] = min(dp[k][q]+abs(a[k][q]-a[i][j]), dp[i][j]);
    }
}
```

上面的$dp(i, j)$表示只考虑$i \times j$这个区域，$(1, 1)$到(i, j)的最短路径。这个方法显然也不行，因为这个动态规划的时间复杂度为$O(n^2m^2)$。

现在思考$(1, 1)$到(i, j)的最短路径，假设某点有较低值a_{q1}满足$a_{q1} \leq a_{11}, a_{q1} \leq a_{ij}$，如果经过此点到$(i, j)$，路径为$a_{11} + a_{ij} - 2a_{q1} \geq |a_{ij} - a_{11}|$；较大值同理。



这就说明去往别的点的最短路径不会包含这些权值很大或者很小的点。如果去往中间点，例如满足$a_{11} \leq a_q \leq a_{ij}$，路径为$a_{ij} - a_q + a_q - a_{11} = a_{ij} - a_{11}$，或者$a_{11} \geq a_q \geq a_{ij}$，这时为$a_{11} - a_{ij}$。

这是惊人的发现，从$(1,1)$去往(i,j)的最小权值就是$|a_{11} - a_{ij}|$，因为中途去别的点都是无意义的。

题目求解

没有必要的中间过程，直接输出$|a_{nm} - a_{11}|$的值。

时间复杂度

时间复杂度$O(T \cdot n \cdot m)$，这其实只有读数据的时间，答案是秒解的。

核心代码

```
for (int i=1;i<=n;i++){
    for (int j=1;j<=m;j++){
        int a; scanf("%d", &a);
        if (i==1&&j==1) r1 = a;
        if (i==n&&j==m) r2 = a;
    }
}
result = abs(r1-r2);
```