20231106 模拟试题解析

1.中位数

【思路点拨】考虑到要用 O(n)的算法，因此不能单纯的枚举左右端点。先分析为什么枚举两端会造成时间的浪费，因为直接枚举两端会遇到很多不可能的情况。如中位数是 4 时，就有可能枚举到 1、4、2、3、5，这样的序列，而这对结果是不会影响的，我们想要的是只统计需要的序列。于是，就要先分析目标序列的性质。不难发现，一个合法的序列必定满足如下性质：

1.序列必定跨越中位数。

2.序列中比中位数大的数的个数一定等于比中位数小的数的个数。

然而，这两个条件不足以用来构造 O(n)的算法，因为数字是散乱分布的，在中位数两侧都有比它大和比它小的数。

但是，根据性质 1，我们可以人为地将合法序列[L,R]拆分为左右两部分[L,M-1]，[M+1,R]；左边比 M 大数为 B1，比 M 小的数为 S1；右边比 M 大的数为 B2，比 M 小的数为 S2。依据性质 2，就可以写出如下等式：B1+B2=S1+S2=>B1-S1=S2-B2；

得到这样的式子就很高兴了。因为左边表示的是中位数左侧的比 M 大的净含量，而右侧表示的是比 M 小的净含量，这两个值都是可以通过一次扫描求出的。

因此，一个扫描算法应运而生：

1.找到中位数的位置 p。

2.从 p 向右扫描到 N，得到每一个点的 S2-B2 的值，并用统计右边的 S2-B2 的值的数量。

3.从 p 向左扫描到 1，得到每一个点的 B1-S1 的是，调用对应的 S2-B2 的数量，累加即可。

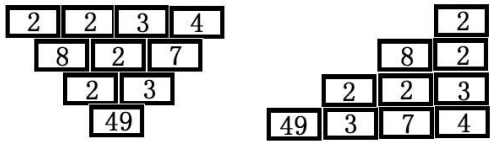
至此，我们就求出了所有的中位数是 M 的序列

2.敲砖块

【思路点拨】首先我们会发现这道题目跟一道经典题目数字三角形很相似，原题是从上到下（只能向左下或右下走）找一条通路，使所得权和最大。本题我们初步构造，设 f[i][j][k]为取第 i 行第 j 列共选了 k 个的最大权值，模仿数字三角形状态转移方程写为：

f[i][j][k]=max{f[i][j][k],f[i-1][j][q]+f[i-1][j+1][p-q+1]+a[i][j]}。经过简单的模拟我们会发现有重复节点；层数越深，重复越多。

我们之所以不能很好地解决问题是因为我们原来以每一个横行为阶段看待问题的角度不符合无后效性。实际上，我们可以换个角度来看问题，按照斜线来划分状态。把原来的图形旋转 90 度，即如下图：



设 f[i][j][k]表示打到第 i 行，总共打了 j 个方块，其中第 i 行打了 k 个方块；

状态转移方程为：f[i][j][k]=max{f[i][j][k],f[i-1][j-k][p]+sum[i][k]}；其中 1<=i<=n,1<=j<=m,1<=k<=I,k-1<=p<=i

sum[i][k]为第 i 行前 k 个数的和。

时间复杂度：$O(n^3m)$

空间复杂度：$O(n^2m)$

【算法优化】：实际上我们在行与行之间状态转移的时候无端浪费了很多时间重复计算上一阶段最大值，因此我们可以对其进一步优化为：

设 g[i][j][k]表示 max{f[i][j][k']},(k'>=k)

f[i][j][k]=max{g[i-1][j-k][k-1]+sum[i][k]}

这样优化后的理论时间复杂度是 $O(n^2m)$。

3.序列合并

【思路点拨】首先把 A 和 B 两个序列分别从小到大进行快排，变成两个有序队列。这样在 A 和 B 中各任取一个数相加可以得到 N^2 个和，可以把这些和看成 n 个有序表:

A[1]+B[1] <= A[1]+B[2] <=…<=A[1]+B[N]

A[2]+B[1] <= A[2]+B[2] <=…<=A[2]+B[N]

……

A[N]+B[1] <= A[N]+B[2] <=…<=A[N]+B[N]

相当于将 N 个有序队列进行合并。首先将这 N 个队列的第一个元素放入堆中。然后每次取出堆中的最小值，如果这个最小值来自于第 k 个队列，那么将第 k 个队列的下一个元素放入堆中，时间复杂度为(NlogN)。

4.最小密度路径

【问题简述】给一个有向无环图，求任两点间距离除以边数的最小值。

【思路点拨】

由于都是求最小，很容易想到和此题类似的一个问题，求任两点间的最短路，能否借鉴 Floyd 算法来解决呢？

本题不同点在于，还要除以一个边数。因为这个除法的缘故，就使得 Floyd 算法的最优子结构性质被破坏，假设存在路径 i→k→j，它的最小密度路径并不一定是 i→k 的最小密度路径加上 k→j 的最小密度路径。

例如：设[A,B]表示路径的权值和为 A，通过了 B 条边。假设从 i→k 存在着两条路径 L1[2,3]以及 L2[8,10]，从 k→j 存在着两条路径 L3[1,2]以及 L4[51,100]，很明显 i→k 的最小密度路径是 L1，k→j 的最小密度路径是 L3，但是 i→k→j 的最小密度路径却是 L1+L4。

那么有否办法能够去掉这个除法的影响呢？

回到原问题，由于是一个有向无环图，一条路径最多只能经过 N-1 条边，于是我们可以对边数进行枚举，即把答案的分母枚举，剩下的就是让答案的分子最小化（答案是权值和/边数），这样就回到我们熟悉的问题——求最短路径。

在 Floyd 的基础上重新划分阶段和定义状态：

第 k 个阶段表示恰好通过 k 条边两点间的最短路，这样最优子结构和无后效性都得到满足（k 的阶段的最优取值一定需要靠之前阶段的最优值，当然也不可能影响到之前阶段的取值了。）

定义状态 f[i][j][k]表示从 i 到 j 恰好经过 k 条边的最短路；

类似 Floyd 的算法得出 DP 方程：

f[i][j][k]=Min{f[i][h][g]+f[h][j][k-g]}。

这个方程是 5 维的，肯定会超时，需要减小维数。考虑在何处重复决策。注意到 f[i][j][k]的选择路径 V1－V2－...－Vk，实际上我们只要找到这里的一个点决策即可，而不需每个点都判断过去。这样就很容易想到在最后一个点进行决策。

f[i][j][k]=Min{f[i][h][k-1]+f[h][j][1]}。