第7场解题报告

visit解题报告

出题人：王泽州

子任务1：-1

只需要判断是否需要往左右走即可。

子任务2：$s = 1, t = 0 || s = n, t = n - 1$

显然是从一个端点走到另一个端点。

子任务3：$n <= 15$

状压即可。$O(n^3 * 2^n)$。

子任务4：$n <= 200$

将经过的点按照顺序写下来我们得到一个排列，我们的答案就是相邻两项的距离和。

设$f[i][0/1][0/1][j][k]$代表前i个点都放完之后，第一个位置前是否能放，最后一个位置后是否能放，有j个位置向左走，有k个区间还可以加入点的最小花费。

在加入$i + 1$是我们只需要枚举它加入的位置左右区间是否还会放更大的点，特殊转移s即可。

$O(n^3)$。

子任务5：$n <= 100000$

考虑贪心的思路。

我们可以只第一步考虑向左走，然后向右走$reverse$一下即可。

我们证明一定是先走完左边再走右边。

情况：a_1, a_2, s, a_4。

若我们$s- > 2- > 4- > 1$，权值和为$a_s + 2 * a_4 - 2 * a_2 - a_1$。

若我们先走右边，权值和为$2 * a_4 - a_s - a_1$。小于上式。

所以我们可以考虑先走完左边，再走右边。

若$t < s$，显然我们依次向左走，在最后一步走到1，然后向右走最优。

否则，我们走完左边，然后还剩下$t - s + 1$次向左走。

考虑将所有右边通过向左走被到达的点设为黑点。

对于一个连续的黑色区间$[l, r]$，我们一定是跳到$r + 1$后依次走到l再到下一个白点。

这样相邻区间的点会被经过三次，但是特殊情况是$r = n - 1$这样只有两次，用一个set扫一边统计答案就行了。

$O(n*logn)$

题目总结：

这个题不算很难，相信大家都可以拿到比较高的分数。

40分是签到的。

70分的dp难点主要在于怎么设计状态。

对于正解的贪心做法需要大家对于策略的分析。

代码写起来比较简单，主要考察思路，是一个很好的题。

Pigeon 解题报告

出题人：张一玎

题目大意

有 n 道题，x和y两人做第 i 题的时间分别是 x_i, y_i. 初始安排时要依次把第 $1, 2, .., n$ 道题放入x或y的待做队列的队尾。

x和y同时开始做自己队列中的题，做完一题后, 如果这题另一个人没做过的, 就加入他的队列中。 如果期间某人列队空了, 那么之后放到他队列里的题全都会被他鸽掉。 问有多少种初始安排方法, 能使A和B都能做完所有题。

$n \le 47, 1 \le x_i, y_i \le 20$

算法1

暴力枚举所有 2^n 种方案，模拟两个人做题的过程，判断是否可行，复杂度 $O(2^n * n)$ ，可以通过Subtask1

算法2

假设x鸽了，那么x至少已经把自己的题做完了，并且y有某道题没有及时给他，这时所有题都加到了y的队列里且y还在做题，y显然不会鸽。得出结论x和y最多只会有一个人鸽。

我们可以分别求出x和y鸽了的方案数，然后用 2^n 减一下就是最终答案。

假设鸽的人是x，设x的队列里的题目为 $a_1, a_2, \dots, a_{n1}$ ，y队列里的题目为 $b_1, b_2, \dots, b_{n2}$ ，设 $sum_a = \sum_{i=1}^{n1} x_{a_i}$ ，这时x鸽的条件为 $sum_a < y_{b_1}$ ，否则y把b1丢过来之后鸽的条件为 $sum_a + x_{b_1} < y_{b_1} + y_{b_2}$ ，依此类推，可得x鸽的条件为：

$$\exists i \in [1, n2], sum_a + (\sum_{j=1}^{i-1} x_{b_j}) - (\sum_{j=1}^{i} y_{b_j}) < 0$$

也就是说左边这个式子对于所有 i 的最小值 <0，x就会鸽，可以利用dp求解。设 $f[i][mn][cur]$ 表示考虑前 i 道题，左边式子所有时刻的最小值为 mn，当前 $sum_a+\sum_{j=1}^{i'}(x_{b_j}-y_{b_j})$ 的值（就是左边式子没有减最后一个 y_{b_i}）为 cur 的方案数，最终答案就是 $f[n][mn<0][cur]$ 的和。转移的时候考虑第 i 道题是放到x的队列里还是y的队列里就行了，具体来说就是这样:

```
f[i+1][mn+x[i+1]][cur+x[i+1]]+=f[i][j]; //放到x的队列里，最小值会受影响
f[i+1][min(mn,cur-y[i+1])][cur+x[i+1]-y[i+1]]+=f[i][j]; //放到y的队列里,用cur更新最小值
```

dp数组较大（mn和cur两维都是 -1000 到 1000），需要滚动数组，时间复杂度 $O(n^3w^2)$，其中 w 表示 x_i,y_i 的最大值（就是20），算一下大概 $4*10^8$ 了（考虑了x和y两个人算两遍、mn和cur两维数组开了两倍这类常数），不过这个上界非常松，加一点常数优化就行了（不过tuoj貌似跑得挺快的，直接这样写也能过。。。）。首先，mn那一维如果 >0，显然没有知道mn具体是多少的必要（如果不通过cur更新，mn的值只会增大），所以 >0 的直接当成0就行了，mn这一维只需要维护 $[-1000,0]$。其次，可以对于每个 $f[i][mn]$ 记录一下dp值不为0的cur的最小和最大值。由于这个dp数组有很多位置都是0，加上这个优化能快不少。

关于Subtask2

这题数据范围本身比较小，设计部分分的空间自然也比较小，在和队友讨论本题的时候队友尝试了直接dp计算合法方案数的做法，复杂度会比正解多出一个 $n*w$，介于这种做法思考和编写的复杂程度都比正解大，这里就不详细写了。如果有人尝试直接dp计算两人都不鸽的方案数，这个部分分应该可以拿到。

总结

这是一道思考难度适中、码量较小的题，考察了选手对dp的掌握程度，做法没有套用任何现成的dp优化套路，而是从题目本身出发，设计合理的dp状态，并通过发掘题目自身的性质进行优化，解法自然，难度适中。数据范围较小（只有47），但是达到了能够支持的上限，而且对解法的时间复杂度不具有提示性，虽然数据范围导致了部分分设置较少，但是也足够区分不同复杂度的做法。

subseqence 解题报告

出题人：袁浩天

子任务1 $n\le 10^5$

将A数列暴力求出，状态$dp(i,j)$表示考虑前i个数以j结尾的本质不同的子序列的个数。其中$dp(i,k)=1$表示空序列

显然：$dp(i,j)=dp(i-1,j),j\neq A_i$

$dp(i,j)=\sum_{l=0}^{k}dp(i-1,l),j=A_i$ 暴力转移即可。

时间复杂度$O(nk)$

子任务2 $n\le 10^{18},k\le 15$

观察可得dp的转移可以看成矩阵乘法，另dp_i为考虑到第i位对应的向量，设$A_i=x$ 转移矩阵则是将单位矩阵的第x(从0编号)行全设置成1， 令该转移矩阵为M_x。

令$F(p,v)$表示对于从M_v开始的k^p个矩阵的乘积。

根据A数组的定义可得转移递推方程：$F(p,v)=\prod_{i=0}^{k-1}F(p-1,(v+i)modk)$

在求得预处理出所有$F(p,v)$后，即可轻松求得最终的答案。

求出F的时间复杂度$O(k^5logn/logk)$ 计算最终结果的时间复杂度$O(k^3logn/logk)$

总的时间复杂度$O(k^5logn/logk)$

子任务3 $n\le 10^{18},k\le 50$

观察到转移方程$F(p,v)=\prod_{i=0}^{k-1}F(p-1,(v+i)modk)$中 $F(p,v)$可以看成是一个前缀积和一个后缀积的积。 所以只需要预处理出$F(p-1)$的前缀积和后缀积即可将计算F的复杂度减少一个k。 总的时间复杂度$O(k^4logn/logk)$

子任务4 $n\le 10^{18},k\le 150$

观察M_i和M_{i+1}可以发现：M_{i+1}其实是在M_i的基础上对左上角的$k*k$的子矩阵作向右和向下的循环位移操作。 令这种操作为 G, 即$G(M_i)=M_{i+1}$

对于满足在第k行和第k列中（从0编号）除了位置(k,k)的值为1，其他元素的值为0 的矩阵A,B ,容易证得以下结论： $G(AB)=G(A)G(B)$ 所以矩阵M_i也满足这种性质，同理$F(p,v)$也满足这种性质。

假设我们已经求得了$F(p,0)$，则进行v次G操作后就能得到$F(p,v)$。

因为$F(p,0)=\prod_{i=0}^{k-1}F(p-1,i)$ 且又有$F(p-1,v)=G(F(p-1,v-1))(v>0)$所以可以使用一个类快速幂计算$F(p,0)$，则计算单个$F(p,0)$ 的复杂度降到$O(k^3logk)$。

总时间复杂度:$O(k^3logn/logk)+O(k^3logklog_kn)=O(k^3logn)$

总结：

此题主要的考点是如何优化dp的转移和对转移矩阵性质的观察，dp的状态设计则比较显然。 此题代码不长，是一道思维难度略高的题。

单场总结

本场比赛题目简短，易于理解。

题目偏向于考察思路，代码简洁。

题目主要考察了dp，贪心，以及矩阵乘法的优化。

题目入手简单，但是进一步的优化需要观察性质或转变方向。

题目部分分较多，即使想不到正解也可以拿到较高的分数。

总之，这是一场大家都可以得到很高分数的比赛。