8.03_信心赛−*solution*

by dst

"题出的好！覆盖知识点广，题目有着切合实际的背景，

解法比较自然。给出题人点赞!"

A.安排妹子

对于30%的数据，枚举。枚举p。

对于100%的数据，二分答案。随着p的增大，$p\ div\ a_1+p\ div\ a_2+p\ div\ a_3+\cdots+p\ div\ a_n=t$的值增大。因此满足单调递增，二分答案$p$。时间复杂度：$O(n\log(a_it/n))$。

B.好感度 up

对于5%的数据，输出0。

对于100%的数据，$DFS/BFS/$并查集。可以先扫一遍边界，空的进去DFS，把连通的点都标记成墙；接着扫一遍所有空间，空的进去DFS，把连通的点都标记成墙，每次DFS时$ans++$。时间复杂度：$O(lmn)$。

C.炒鸡矿工

一．预处理

令$n=n+1,v_1=c,s_1=p$，v_i表示升到第i级单次挖矿增加的重量，w_i表示升到第i级的代价，s_i表示第i级单次挖矿的时间。预处理出v_i的前缀和sv_i，则sv_i表示第i级单次挖矿的重量。

二．对于80%的数据

$f_{i,j,k}$表示总时间在第$i(0\le i\le t)$分钟，等级为$j(1\le j\le n)$级，单次挖矿剩余$k(s_j>k\ge 0)$分钟时，dst拥有的最大金矿重量。以时间（第i分钟）作为状态。

继承：$f_{i,j,k}=f_{i-1,j,(k+1)\%s_j}$。特殊地，当$k=0$时，$f_{i,j,k}=f_{i-1,j,1\%s_j}+sv_j$。

升级：$f_{i,j,k}=\max\{f_{i,j,0},f_{i,j-1,0}-w_j\}(k=0且f_{i,j-1,k}\ge w_j)$。由于可以无限升级，根据完全背包思想，$j$应当正扫。

时间复杂度：$O(tns_i)$。

三．对于100%的数据

优化掉第三维。$f_{i,j}$表示总时间在第$i(0\le i\le t)$分钟且单次挖矿剩余0分钟，等级为$j(1\le j\le n)$级时，dst拥有的最大金矿重量。

由于在一次挖矿结束时收矿，所以对于任意整数$k(k>0)$，在同一时刻同一

等级，单次挖矿剩余x分钟时dst拥有的最大金矿重量一定大于$x+k$分钟。那么单次挖矿剩余0分钟一定是最优的，即$\max\{f_{t,j}(1\le j\le n)\}$一定为最优的答案。

"只能在一次挖矿开始前进行升级"，等价于可以在任何时间升级，但只能在下一次挖矿开始后体现升级效果。因此转移方程可以分解如下：

继承：$f_{i,j}=f_{i-1,j}(i\ge 1)$。收矿：$f_{i,j}=f_{i-s_j,j}+sv_j(i\ge s_j)$。

升级：$f_{i,j}=f_{i,j-1}-w_j(f_{i,j-1}\ge w_j)$。所以转移方程为：

$f_{i,j}=\max\{f_{i-1,j}(i\ge 1),f_{i-s_j,j}+sv_j(i\ge s_j),f_{i,j-1}-w_j(f_{i,j-1}\ge w_j)\}$。

时间复杂度：$O(tn)$。

D . 游戏大师

一．满分

第一问。我们利用容斥原理，观察后不难发现：对于任意i，$(a_i>a_j$或$b_i>b_j$的方案数$)=(a_i>a_j$的方案数$)+(b_i>b_j$的方案数$)-(a_i>a_j$且$b_i>b_j$的方案数$)$。对于$(a_i>a_j$的方案数$)$，$(b_i>b_j$的方案数$)$，只需要以a_i作为关键字或b_i作为关键字进行排序，排序后的位置-1即为方案数。对于$(a_i>a_j$且$b_i>b_j$的方案数$)$，先以a_i作为第一关键字排序，然后按顺序将(下标b_i'，值1)丢入树状数组维护前缀和，并查询下标b_i'的前缀和作为答案。我们考虑当前的状态，先前丢入树状数组中的a指标必然小于当前，即满足$a_i>a_j$；由于查询的是前缀和，因此满足$b_i>b_j$。

第二问。仍然是树状数组维护前缀和，但不需要利用容斥原理，而需尺取法$(two\ pointers)$。先以a_i作为第一关键字排序。我们定义i下标为右指针，j下标为左指针。每次先将i下标右移一个单位，然后将j下标不断右移，直到$a_i\ge a_j\times 2$不成立，并在右移的同时将(下标$b_j'\times 2$，值1)丢入树状数组，最后对于每个i下标，查询下标b_i'的前缀和作为答案。当然，也可以用二分查找代替尺取法，这里不再赘述。

由于$a_i,b_i\le 10^9$，所以需要进行离散化，离散化时应将$a_i,b_i,a_i\times 2,b_i\times 2$都计入。

时间复杂度$O(n\log n)$。空间复杂度$O(n)$。

二．部分分

对于$n\le 10^3$的数据，可以$O(n^2)$暴力。

对于$a_i=b_i$的数据，可以尺取法$O(n)$或二分法$O(n\log n)$。

对于$a_i,b_i\le n$的数据，不需要离散化。