

信息学奥林匹克联赛(NOIP)模拟赛

提高组 思路提示

(请选手务必仔细阅读本页内容)

一. 题目概况

题目名称	序列	数字	背包
可执行文件名	sequence	number	bag
输入文件	sequence.in	number.in	bag.in
输出文件	sequence.out	number.out	bag.out
每个测试点时限	1s	1s	1s
内存限制	128M	128M	256M
测试点数目	10	10	10
每个测试点分值	10	10	10
结果比较方式	忽略多余的空格和文件尾的空行		
题目类型	传统	传统	传统

注意事项:

- 1、文件名（程序名和输入输出文件名）必须使用英文小写。
- 2、C/C++中函数 `main()` 的返回值类型必须是 `int`，程序正常结束时的返回值必须是 0。

1. 序列

(sequence.cpp/c)

【问题描述】

给出一个长度为 n 的整数序列，要求删掉一个连续子串后序列中没有相同元素，请问至少要删掉多长的子串？

【思路】

对于 20% 的数据， $1 \leq n \leq 100$

暴力枚举

对于 100% 的数据， $1 \leq n \leq 1000$

若删除长度为 x 的子串后序列中没有相同元素，那么一定有至少一个长度为 $x+1$ 的子串，删除它后序列中也没有相同元素

2. 数字

(number.cpp/c)

【问题描述】

给出两个整数 n, m

求 1 到 n 的所有整数中，能被 m 整除的整数的个位数字之和

【思路】

对于 20% 的数据 $m, n \leq 100000$ 。

暴力枚举

对于 100% 的数据 $m, n \leq 10^{16}$ ， $T \leq 100$

不妨找找规律？

3. 背包

(bag.cpp/c)

【问题描述】

有一个背包，和 n 个物品，要把某些物品彼此压着放在背包里。

背包最大承重为 S

已知第 i 个物品放上去的时间 $in[i]$ ，拿走的时间 $out[i]$ ，重量 $w[i]$ ，承重 $s[i]$ ，价值 $v[i]$ ， $in[i] < out[i]$

对于同一时刻，有多个物品进出的话，顺序任意

如果将一个物品放入背包，必须满足以下要求：

- 1) 在 in 时间进，放在最上面
- 2) out 时间出，出的时候他在最上面，并且得到价值 v
- 3) 当他在背包内时，他上方的物品的重量必须时刻小于等于 $s[i]$

它也可以不放入背包，则得不到价值 v

问你可以得到的最大价值

【数据范围及约定】

对于 20% 的数据， $n \leq 10$

对于额外 20% 的数据，保证所有 $in[i]$ 相同且 $out[i]$ 相同

对于额外 20% 的数据，保证所有 $in[i]$ 相同

对于 100% 的数据， $n \leq 500$ ， $S \leq 1000$ ，

$0 \leq in[i] < out[i] < 2n$ ， $w[i] \leq 1000$ ， $s[i] \leq 1000$ ， $v[i] \leq 1000000$

【思路】

先把所有物品按照拿走的时间从小到大排序，拿走的时间相同就按照放上去的时间从大到小。那么一件物品上方的物品就一定会在它的前面。

考虑 dp ，设 $f[i][j]$ 表示 i 以及 i 上面物品在所有时刻中最大重量为 j 时的最大收益。

转移的时候，我们需要枚举所有 i 上面的物品，维护一个 $g[i]$ 表示时刻 i 之前物品的最大收益是多少。然后直接转移就好了。