

## 选择题(7')

---

1、2 小题均为时间复杂度相关的问题，难度不大。

第 3 题为 Set Cover 问题的近似比，课件上有推导，答案为  $\log n$ 。如果记得思路现推也不难。如果不记得思路且大抄上没有那我祝你好运。

## T1(19')

---

计算  $A[1], \dots, A[n]$  中恰为  $B[1], \dots, B[m]$  的子序列数量。要求时间复杂度  $O(nm)$ 。

**Sol:**

定义  $f(i, j)$  表示  $A[1], \dots, A[i]$  中能与  $B[1], \dots, B[j]$  匹配的子序列数量。

转移方程为

$$f(i, j) = \begin{cases} f(i-1, j) & \text{if } A[i] \neq B[j] \\ f(i-1, j) + f(i-1, j-1) & \text{if } A[i] = B[j] \end{cases}$$

时间复杂度  $O(nm)$ 。

本题难度个人认为低于实验和作业中涉及的动态规划问题。

## T2(19')

---

给定序列  $A[1], \dots, A[n]$ ，求区间和为 0 的区间数量。要求时间复杂度  $O(n)$ 。

**Sol:**

计算前缀和  $s[i] = \sum_{j=1}^i A[j]$ ，那么如果  $s[i] = s[j]$ ，则说明  $A[i+1], \dots, A[j]$  为区间和为 0 的一个区间。

利用哈希表维护即可做到  $O(1)$  的查询和插入操作，顺着扫一遍即可。

需要注意的是， $s[0] = 0$  同样需要被插入到哈希表中。如果没有注意到这一点会被扣 2 分。

(本题与 Lab2 T2 相似度极高，独立完成实验的同学应该都能做出。而且老师考试时提示了数字可能很大，需要使用哈希表。这应该是本场考试最简单的题目。)

## T3(19')

---

给定  $n$  个任务，每个时间最多只能完成 1 个任务，每个任务的截止时间为  $t_i$ ，价值为  $w_i$ ，最多能够完成  $k$  个任务，请求出能够获得的最大价值。

**Sol:**

本题没有要求时间复杂度，所以做法比较多。

我考场上写的做法是将任务按照价值排序，然后贪心地从高往低将每个任务安排在能完成的最晚时刻，不难证明这样的策略是正确的。

暴力的执行的时间复杂度是  $O(nt)$ ，可以通过线段树上二分将  $t$  优化成  $\log t$ ，不过既然没有要求时间复杂度也不用这么麻烦。

另一种做法是，维护一个大小不超过  $k$  的小根堆，然后按照时间顺序扫描每个任务。如果当前时间  $t_i$  小于堆的大小则直接插入。如果等于（说明此时前  $t_i$  的时间已经被安排满了）则与堆顶进行比较，取较大值保留。最终堆里的任务就是所要完成的任务。时间复杂度为  $O(n \log n)$ 。

本题应该很经典了，而且没有要求时间复杂度，且难度（想写低于作业来着，但仔细一想考前的作业里除了排队接水和近似比为 2 的 01 背包之外并没有别的贪心题目，所以感觉并不成立）其实并不算大。

## T4(19')

---

$N$  为 2 的整数次幂， $x$  为一个  $N \times N$  的矩阵，定义二维傅里叶变换：

$$\hat{x}(f_1, f_2) = \sum_{t_1=0}^{N-1} \sum_{t_2=0}^{N-1} x(t_1, t_2) \cdot \exp\left\{\frac{2\pi i(t_1 f_1 + t_2 f_2)}{N}\right\}$$

设计算法快速计算二维傅里叶变换。

### Sol:

写成矩阵的形式，可以发现二维傅里叶变换本质上就是对  $x$  左右各乘一个 Vandermonde 矩阵，即

$$\hat{x} = V_N x V_N$$

可以通过定义验证。

于是先对  $x$  的每一列做一维的 FFT，再对所得矩阵的每一行做一维的 FFT 即可。时间复杂度  $O(n^2 \log n)$ 。

本题需要一些观察和猜想，且了解 FFT 本身而不是单纯记个代码。Hw3 中的 Hadamard 变换（也是去年的期中考试题）与其具有一定的相似性。且助教在习题课上也讲了 Hadamard 变换从矩阵的角度如何看待，对做出本题应该有所帮助。

## T5(17')

---

计算机中浮点数时常出现误差，所以经常采用分数来存储浮点数。给定  $n$  个形如  $a[i]/b[i]$  的分数，满足  $1 \leq a[i], b[i] \leq n^3$ ，设计一个  $O(n)$  的算法进行排序。

### Sol:

观察到任意两个分数的差最大为  $\frac{n^3-2}{n^3-1} - \frac{n^3-1}{n^3} > \frac{1}{n^6}$ 。

也就是任意两个分数，要么相等，要么乘以  $n^6$  之后会有一个不小于 1 的 gap。那么此时再取整也会保留一个不小于 1 的 gap，也就是说将所有分数都乘  $n^6$  之后就被离散化到了  $[n^3, n^9]$  上，且满足原有的大小关系。

而且由于值域的特殊性，我们执行以  $n$  为基底的基数排序即可做到  $O(6 \times (n + n)) = O(n)$ 。

本题在思维上的跳跃性比较大，我自己考场上并没有做出来（其实下了考场也没做出来，是问助教才会的）。