《算法设计与分析》期末复习

考试提示

- 不考附加题

时间复杂度

有递归表达式$T(n)=aT\left(\frac{n}{b}\right)+O\left(n^d\right)$for $a>1,b>1$ and $d>0$, then $T(n)$ can be bounded by:

- (1) If $d<\log_b a$, then $T(n)=O\left(n^{\log_b a}\right)$;
- (2) If $d=\log_b a$, then $T(n)=O\left(n^{\log_b a}\log n\right)$;
- (3) If $d>\log_b a$, then $T(n)=O\left(n^d\right)$.

作业题

Divide And Conquer

1.Longest Balanced Substring

对于字符串按不平衡（只有大写或小写的）的字符进行分割，然后递归处理这些子字符串。

2. Cutting Bamboo Poles

二分查找进行搜索

3.Multiple Calculations

将每个运算符作为分割，递归计算运算符的左右表达式。然后合并在一起。

时间复杂度计算结果$>2^n$，每次递归合并左右部分的时间复杂度为$O(m^2)$，最后时间复杂度是$O(m^2 2^n)$

4.N-sum

$p(x)=\prod_{j=1}^{n}x^{B[i_j]}=x^{\sum_{j=1}^{n}B[i_j]}$

则$\sum_{j=1}^{n}B[i_j]$都在p^n指数中，只需要计算出p^n，然后验证第m位是否存在就可以。

利用FFT来实现大数的乘法，FFT的时间复杂度是$O(nlogn)$，总共要做$n-1$次FFT。

A.Nearest Point

- 直接暴力可以求解
- 用分治的方法求解，将点按x坐标排序，然后分别处理左右两边。左右合并时，在中间部分，设$delta$为左右两边中更小的结果，那便只有处于距中间线 $delta$以内的点对才会可能比$delta$更小。搜索中间部分时，可按 坐标排序并依据进行剪枝，此排序利用分治过程进行归并排序以减少运算。

B.A unique permutation

l是一个逆等差数列，则$2l$和$2l+1$也是。并且将$2l$和$2l+1$拼接在一起也是逆等差数列。

C.Lost items

暴力求解时间复杂度是$O(N*2N)$，太大了，过不了，于是考虑将其分成两部分每部分各有$\frac{N}{2}$个物品，分别计算各自的总量集合。然后将两个进行合并，枚举A集合中所有重量，然后查找（目标重量 $-$ W_A）是否在集合B中

Dynamic Programming

1. Step Problem

$dp[n]=dp[n-1]+dp[n-2]$

2. Buy!

01背包:

$dp[i][j]=max(dp[i-1][j],dp[i-1][j-w_i]+v_i)$

完全背包

$dp[i][j]=max(dp[i-1][j],dp[i][j-w_i]+v_i)$

区别在于$dp[i][j-w_i]+v_i$，原因是完全背包可以选取i物品任意次。

3.Counting

可以将问题考虑为将S中的若干个元素拼成长度为n的数组，最终完美数组的个数等于**拼接的方法数和长度为S_k数组个数**控制

$N(S_k)=(m-k+1)^{S_k}-(m-k)^{S_k}$

$opt[n]=\Sigma_{i=1}^{m}N(S_i)opt(n-S_i)$

A. Optimal Display Rental

$dp[i][l]=max(dp[i][l],dp[j][l-1]+c_i)$

B. Sheep Transport Problem

$dp[j] = min(dp[j], dp[j-i] + sum[i])$

C. Ant Foraging

要将其转化为不具有后效性的问题（就要增加DP的维度），这里转化成两只蚂蚁同时从起点出发到达终点，并且互相不能走对方的路径。求最大化两条路径的和。

$$opt[x_1][y_1][x_2] = max \begin{cases} opt[x_1-1][y_1][x_2] \\ opt[x_1-1][y_1][x_2-1] \\ opt[x_1][y_1-1][x_2] \\ opt[x_1][y_1-1][x_2-1] \end{cases}$$

Greedy

1. Distributing Candy Game

根据题目的要求，小朋友1和小朋友2位置交换的必要条件是：小朋友2在小朋友1前面时所获得的糖果数应更小。若小朋友1在前面，则他们的糖果数为$\frac{a_0}{b_1}$和$\frac{a_0 a_1}{b_2}$；若小朋友2在前面，则 他们的糖果数为$\frac{a_0}{b_2}$和$\frac{a_0 a_2}{b_1}$。两个小朋友要交换的条件是$max(\frac{a_0}{b_1}, \frac{a_0 a_1}{b_2}) > max(\frac{a_0}{b_2}, \frac{a_0 a_2}{b_1})$，分类讨论可以得到$a_1 b_1 \geq a_2 b_2$，所以将孩子们按照$a \cdot b$从小到大排列即可满足条件。

2. Array Partition

排序之后选择偶数位置的数求和

3. Delete Number Game

从n位整数中选择k位进行删除，即从n位整数中选择$n-m+1$位使得新的整数足够小，只需要在选择每一位时都选择最小的数字。但是要保证后续有足够的位置选择剩余的数字。

A. Sorrowful Cows

可以计算出每个有牛格子之间的间距，按降序排序，板子要避免封住间隔较大的格子。

B. Sick Cows

先求出传染天数：$min(min(\text{中间的牛群的最多传染天数}), min(\text{边界牛群最多传染天数}))$

求出0号牛的个数：$\lceil \frac{x}{2m+1} \rceil$

C. Minimum Number of Straights

```
N = int(input())
N_cards = [int(input()) for i in range(N)]

res = N_cards[0]
for i in range(N-1):
    if N_cards[i+1]>N_cards[i]:
        res += N_cards[i+1] - N_cards[i]
```

Linear Programming

线性规划与对偶问题

- 原问题是不等式约束
  - 原问题

$$\begin{aligned} min \quad & c^T x \\ s.t. \quad & Ax \geq b \\ & x \geq 0 \end{aligned}$$

  - 对偶问题

$$\begin{aligned} max \quad & b^T y \\ s.t. \quad & A^T y \leq c \\ & y \geq 0 \end{aligned}$$

- 原问题是等式约束
  - 原问题

$$\begin{aligned} min \quad & c^T x \\ s.t. \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

  - 对偶问题

$$\begin{aligned} max \quad & b^T y \\ s.t. \quad & A^T y \leq c \end{aligned}$$

1. Maximize profit

$\text{Maximize } Z = 3x_1 + 5x_2$

s.t.:
$2x_1 + x_2 \leq 6$
$x_1 + 3x_2 \leq 9$
$x_1 \geq 0, x_2 \geq 0$

2. Shipping goods

$\text{Minimize} \quad 8x_{1A} + 6x_{1B} + 10x_{1C} + 9x_{1D} + 7x_{2A} + 5x_{2B} + 8x_{2C} + 7x_{2D} + 6x_{3A} + 7x_{3B} + 7x_{3C} + 8x_{3D}$

s.t.

$x_{1A} + x_{1B} + x_{1C} + x_{1D} \leq 50$

$x_{2A} + x_{2B} + x_{2C} + x_{2D} \leq 60$

$x_{3A} + x_{3B} + x_{3C} + x_{3D} \leq 40$

$x_{1A} + x_{2A} + x_{3A} = 40$

$x_{1B} + x_{2B} + x_{3B} = 50$

$x_{1C} + x_{2C} + x_{3C} = 40$

$x_{1D} + x_{2D} + x_{3D} = 20$

$x_{ij} \geq 0$, 对于所有 i 和 j

3. Production

$$\text{Maximize} \quad 40,000x_1 + 50,000x_2 + 60,000x_3 - 1,000,000y_1 - 1,500,000y_2 - 2,000,000y_3$$

s.t.

$$2x_1 + 4x_2 + 8x_3 \leq 500$$

$$2x_1 + 3x_2 + 4x_3 \leq 100$$

$$3x_1 + 6x_2 + 9x_3 \leq 300$$

$$0 \leq x_1 \leq My_1, \quad 0 \leq x_2 \leq My_2, \quad 0 \leq x_3 \leq My_3$$

$$y_1, y_2, y_3 \in \{0, 1\}$$

A. Card Selection

可以写出线性函数的目标和约束

B. ILP Feasibility

整数规划问题，虽然可以用求解整数规划的方法解决（分支界定法等），但这样会很复杂，而且时间可能过不去，要考虑其他方法。对于这种简单的约束可以考虑动态规划。

动态规划，以需要满足的条件为index，以需要优化的值作为value

$dp[V] = min(dp[V - c_i] \dots) + 1$

C. Production Line Optimization