

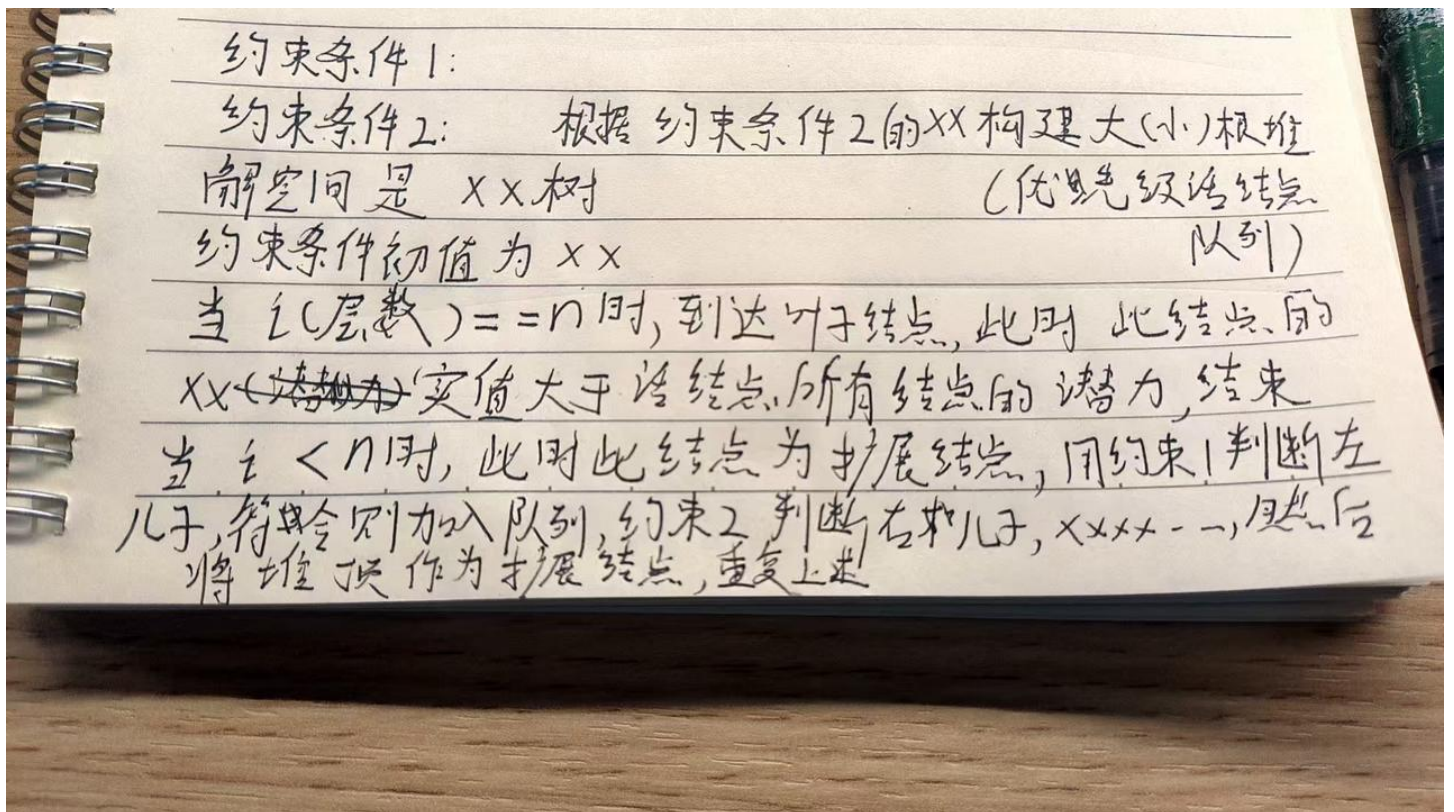
算法复习

期末成绩:93 过程考核成绩:94

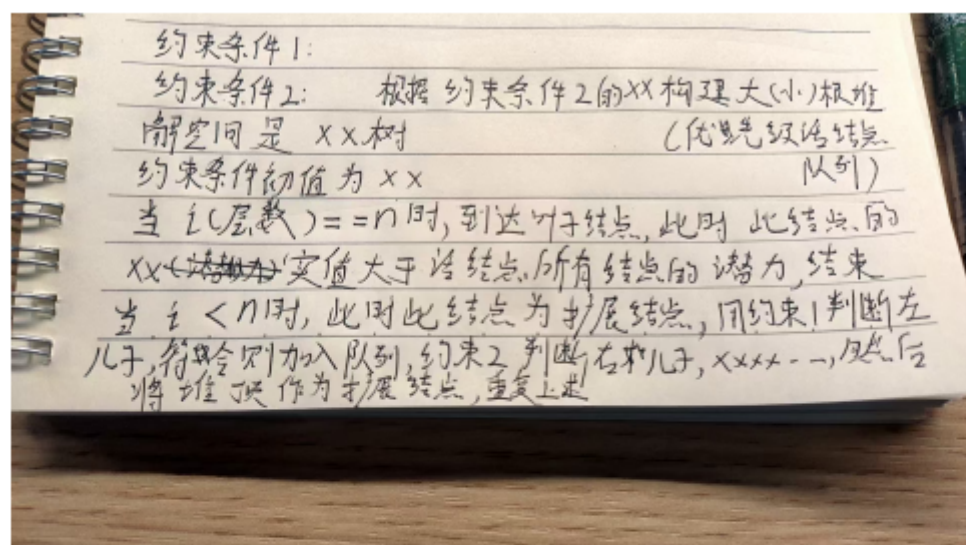
纯文科课程，没多少代码，突击1.5天可以90+，就按照我下面总结的话术背诵即可

自总结分支限界算法和回溯算法的答题背诵模板

分支限界：



这是回溯和分支限界



lt

12-10 18:20:15

我打算考试就这么写了

lt

12-10 18:29:01

$i=n$ 时到达叶子节点，此时若叶子节点的当前值更小，更新best，否则回溯

$i < n$ 是说明节点在解空间里面，不满足则约束条件剪枝，若满足，如果当前值更小，更新best，然后按照深度优先搜索继续求解

lt

12-10 18:29:14

把最后两句替换成这两句就行

lt

12-10 18:29:38

可能 n 有个 $+1$ 或者 -1 ，但这都无伤大雅了

下面是一点点概念

第一章

- **算法：是满足下述性质的指令序列。**
 - 输入：有零个或多个外部量作为算法的输入。
 - 输出：算法产生至少一个量作为输出。
 - 确定性：组成算法的每条指令清晰、无歧义。
 - 有限性：算法中每条指令的执行次数有限，执行每条指令的时间也有限。
 - 程序可以不满足算法的性质(4)即有限性。
- **算法与数据结构的关系**
 - 不了解施加于数据上的算法就无法决定如何构造数据，可以说算法是数据结构的灵魂；
 - 反之算法的结构和选择又常常在很大程度上依赖于数据结构，数据结构则是算法的基础。
- **算法 + 数据结构 = 程序**
 - **O的定义**：如果存在正的常数C和自然数 $N(0)$ ，使得当 $N \geq N(0)$ 时有 $f(N) \leq Cg(N)$ ，则称函数 $f(N)$ 当N充分大时有界，且 $g(N)$ 是它的一个上界，记为 $f(N) = O(g(N))$ 。即 $f(N)$ 的阶不高于 $g(N)$ 的阶。

第二章：

分治法的**基本思想**是，将一个难以直接解决的大问题，分割成一些规模较小的相同问题，以便各个击破，分而治之。

- 直接或间接地调用自身的算法称为**递归算法**。
- 用函数自身给出定义的函数称为**递归函数**。

边界条件与递归方程是递归函数的二个要素，递归函数只有具备了这两个要素，才能在有限次计算后得出结果

分治法所能解决的问题一般具有以下几个特征：

- 该问题的规模缩小到一定的程度就可以容易地解决；
 - 该问题可以分解为若干个规模较小的相同问题，即该问题具有最优子结构性质
 - 利用该问题分解出的子问题的解可以合并为该问题的解；
 - 该问题所分解出的各个子问题是相互独立的，即子问题之间不包含公共的子问题。
-

1. 这条特征涉及到分治法的效率，如果各子问题是**不独立**的，则分治法要做许多不必要的工作，重复地解公共的子问题，此时虽然也可用分治法，但一般用**动态规划**较好。
2. **区别2**：分治算法是自顶向下，动态规划是自底向上

分治法的**基本步骤**

divide-and-conquer(P)

```
{  
  if (|P| ≤ n0) adhoc(P); //解决小规模的问题  
  divide P into smaller subinstances P1,P2,...,Pk; //分解问题  
  for (i=1,i≤k,i++)  
    yi=divide-and-conquer(Pi); //递归的解各子问题  
  return merge(y1,...,yk); //将各子问题的解合并为原问题的解  
}
```

人们从大量实践中发现，在用分治法设计算法时，最好使子问题的规模大致相同。即将一个问题分成大小相等的k个子问题的处理方法是行之有效的。这种使子问题规模大致相等的做法是出自一种**平衡 (balancing)**子问题的思想，它几乎总是比子问题规模不等的做法要好。

这里分享一下22级算法考试：

- 1.本人能力局限，题目不难，但我写的太慢，最后也没完全写完。
- 2.考试题目来源：

目 录

O(n) 计算

2.6 2.8 2.9 2.11

3.1 3.3 3.10

4.1 4.2 4.3 4.5 4.6

5.2 5.4 5.5 5.6 5.7 5.9

6.5 6.6 6.7

7章 all

算法考点

4x4 简答.

2x10 编程

8x8 综合

3.只有编程题不会考原题，只有编程题必须写代码，综合题和简答题可以文字描述。

QCGB