|  |
| --- |
| 有很多同学纷纷表示看到专业课就头大，来问我专业课怎么复习？ 别急，听我慢慢给您道来～  先说下大致的复习节奏，只要节奏对了至少不会出现大的纰漏。  复习规划  7-8月暑假：       课本一定要看！！！820难度不大，却以阴险著称。考的很细，经常考察基本概念，没错，就是让你背概念，这是其一；其二是偶尔会出些课本犄角旮旯里的知识，让你防不胜防。所以，课本至少要过上一遍。不要觉得这是浪费时间，这也是为后面做基础，子曰磨刀不误砍柴工嘛。       把《计算机操作系统》（汤小丹版）和《数据结构》（严蔚敏版）课本过一遍，看完课本上一章节的内容，做王道上相应章节的题。《数据结构》（严版）这本书上算法的代码有难度，如果看不懂的可以放一放，以王道书上的算法为准。底子薄弱的同学，第一遍可以只做王道上的客观题和简单的大题。  9月-10月：        以王道书为基本，再做一遍，做的时候主攻大题、重难点以及自己薄弱的地方。期间发现有疑问的地方，要在课本上找到相应的章节再次强化理解。 这一遍 至少 要达到看到主流的题，就能想到相应的解题方法。  10-11月：        对照820考纲速看王道书第三遍，对可能的考点重点理解，不能有遗漏的地方。注意：820考纲和王道书覆盖的知识点略有不同，需要把王道上不包括的知识点补上，有些820不考的知识点可以不再花费时间去复习。  11月：        820历年真题，做一套总结一套，注意真题考点的侧重点以及自己还存在疑问的地方。  12月：        根据真题结合王道书，自己总结出所有的考点和可能考察的算法。然后对这些内容不断强化记忆，直至考试。  大致安排如此，可以根据时间和自己的情况做调整。   复习方法 \*\*\*操作系统\*\*\*       820在操作系统方面有几个特点，重基础重概念考察非常细致。有些年份会出好几道简答题，让你解释某一概念，甚至是详述某一名词是什么意思。复习的时候一定要留意，820的出题风格、题型是和王道书上的题型有一些差别的，我第一次看真题的时候也是很崩溃。 例如16年真题： 三、简答题（ 4 题，共 21 分） 1. PCB 的主要存储内容是什么？为什么说 PCB 是进程存在的唯一标志？（ 6 分） 2. 什么是虚拟存储器？如何实现页式虚拟存储器？（ 5 分） 3. 什么是设备的独立性，应如何实现？（ 5 分） 4. 文件物理结构是指一个文件在外存上的存储组织形式，那么何谓文件的混合索引结构？ 其主要优点是什么？（ 5 分）         这样的题型在统考中是见不到，那么如何应对？首先要理解和总结，然后去背！必须刻意的记忆。        类似这样的简答题可谓是极其的阴险，仅达到理解、会做题这一程度来做这种看似送分的简答题时，往往会因为答的不准确而失分。这时就需要你自己去总结了。对照820的考试大纲，上面都写有会考察的知识点和概念，把涉及到内容都要单独总结出来。总结的时候要一块一块的总结，比如要总结文件的物理结构，你就要把所有的物理结构总结在一起，分别写出不同物理结构的特性、优缺点。这样成块总结，对比着去理解记忆会更容易些。        在前两轮复习的时候以做题为主，能把知识点理解就行，等真题做完一遍后，根据真题和考纲再来总结。总结好后，在考试前冲刺复习阶段，花一些时间把这些总结的内容都背下来。 （其实和期末考试考前突击一个套路） 多说一句，做真题的时候要注意经常出题的点，比如进程调度、进程同步、页面置换、文件索引这几个爱出题的点要多做些题把做题的方法搞熟练。   \*\*\*数据结构\*\*\*        数据结构的难点应该就是算法了，有同学说算法能看懂，但是让自己写却写不出来。其实这还是不熟练所造成的。王道书上的算法都比较精炼，理解起来应该问题是不大的。遇到比较复杂的算法，可以画出流程图和数据栈一步一步的模拟运行过程，这样可以加深理解。        算法这部分其实是有一些技巧的。        算法主要分线性表、树、图三个部分，中间又有和查找、排序相结合的部分。820中线性表和树考察的比较多；图的算法只考过应用，基本上没有考过编程题；主要是因为图的算法太过复杂，难度高且不易批改。在复习的后期，如果图的算法是在写不出来，可以做一些取舍。（这只是不得已的做法，有能力的同学千万不能偷懒，万一呢？对不对）        在线性表中，主要的查找和排序算法都要总结出来。链表中的常用方法，比如头插法、尾插法、逆置法、归并法和双指针法，这些常见的算法也要总结出来。在树中，递归的遍历算法，非递归的遍历算法这是最最基本的算法都要总结出来，在总结的时候要讲方法。比如树的递归遍历算法： void order(BiTree T){     if(T!=NULL){         ①         order(T->lchild);         ②         order(T->rchild);         ③     } }       对遍历操作的思想有所理解后，你会发现所有遍历方法对结点的访问顺序是一致的，仅仅是对结点的读操作的相对位置不同。根据这个特点，我们就能得到上面这个算法的框架，然后在不同的位置上加上visit（T）操作，就会分别编程前中后序遍历算法。如果你以为总结到这种程度就够了，那只能说你简直太年轻！       随着做题数量的增加，你会发现基本上所有关于树的算法题都是基于遍历的操作的。举个例子吧，写出交换所有左右子树的递归算法。首先你要选择一个适合的访问顺序，这里先序遍历就能实现，那么我们就在①的位置写入交换左右子树的代码即可。连然后都不需要，题就做完了！这就是总结的目的，要把不同算法的本质找到总结出来，然后遇到这一类的题时就不断的剖析，看看和哪个算法框架相似，找到后就不断在框架里添加满足题意的操作，直至能实现为止。（即使是做错了，只要框架和思路没错都是有分的）         这是一个做题思路，还有另外一个方法，那就是背！没错，把王道书正文中的代码全部达到熟练默写的程度，习题中自己完全没有思路的算法也都背下来。只要背熟，再次遇到相似的题，即使写不对，也能写个七七八八的。还有背的熟练，才能在考场上又快又准确的写出相应的算法，这是得高分必不可少的条件。背默在加上之前的总结，你会发现编程题好像也没有那么的难了～  愿学弟学妹们都能实现自己的梦想！加油哦～ |