

灰灰考研

2021 复旦大学

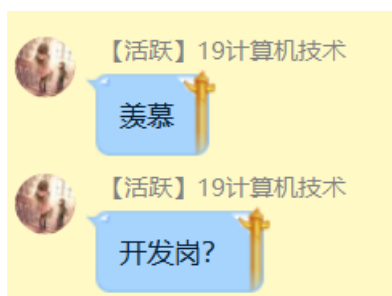
计算机复试手册 V1.0



【皮皮灰】
免费提供，严禁销售

灰灰考研

2021 复试时间节点【预测】	3
复试需要携带的物品	4
复试科目相关.....	5
上机	7
2019 年复旦大学上机题	8
2018 年复旦大学上机题	13
2017 年复旦大学上机题	14
2016 年复旦大学上机题	15
2015 年复旦大学上机题	16
往年年复旦大学上机题.....	17
复旦 OJ 题目	20
面试	45
历年复试细则.....	73
2019 年复旦大学计算机科学技术学院硕士研究生招生复试实施细则.....	73
2019 年复旦大学软件学院全日制硕士研究生招生复试实施细则	78
2019 年复旦大学软件学院软件工程（非全日制）专业学位硕士研究生招生复试实施 细则	81
复试经验	85
2017【一】	85
2017【二】	86
2018【一】	86
2018【二】	87
2018【三】	88



2021 复试时间节点【预测】

初试成绩查询时间

预计 2021 年 2 月 27 日【元宵节后】

2019 年查询时间 2019 年 2 月 15 日

2018 年查询时间 2018 年 2 月 3 日

2017 年查询时间 2017 年 2 月 16 日

国家线公布时间

2021 考研国家线预计 2021 年 3 月中旬公布

2019 考研国家线—— 2019 年 3 月 15 日公布（当年春节是 1 月 25 日）

2018 考研国家线—— 2018 年 3 月 16 日公布（当年春节是 1 月 28 日）

2017 考研国家线—— 2017 年 3 月 15 日公布（当年春节是 2 月 16 日）

复旦大学计算机复试时间

2019 年计算机学院复试细则公布：2019 年 3 月 11 日

2019 年计算机学院复试时间：2019 年 3 月 18 日-19 日

2019 年软件学院复试细则公布：2019 年 3 月 12 日

2019 年软件学院复试时间：2019 年 3 月 21 日-22 日

历年调剂系统开放时间

2019 年调剂系统：3 月 20 日-4 月 30 日

2018 年调剂系统：3 月 23 日-4 月 30 日

2017 年调剂系统：3 月 18 日-4 月 30 日

复试需要携带的物品

按照《复旦大学 2019 年硕士考生报考资格审查要求》，考生应携带以下材料参加

复试：

- (1) 考生的准考证和有效身份证件原件。
- (2) 应届生的学生证原件。
- (3) 往届生的学历证书原件和教育部学历证书电子注册备案表（或学历认证报告），学位证书原件（报考对学位有要求专业的考生）。
- (4) 持在境外获得学历/学位的考生，须提交教育部留学服务中心出具的认证报告（境外学校在读本科生，不能以应届生身份报考）。
- (5) 对于同等学力考生，须查验授课学校教务部门提供的 8 门相关专业本科课程考试成绩单原件。
- (6) 对于在读研究生，须查验培养单位出具的同意报考证明；考生还需于拟录取前提供注销原学籍证明。
- (7) 对于在 2019 年 9 月 1 日前可取得国家承认本科毕业证书的自学考试和网络教育本科生，须查验颁发毕业证书的省级高等教育自学考试办公室或网络教育高校出具的相关证明。
- (8) 对于少数民族骨干计划考生，须查验是否具有《报考 2019 年少数民族高层次骨干人才计划硕士研究生考生登记表》。没有《登记表》的，不能列入少数民族骨干计划。
- (9) 对于“退役大学生士兵”专项计划考生，须查验是否具有《入伍批准书》和《退出现役证》。

在资格审查时，考生须携带上述材料原件和复印件各一套，复印件由院系留存。经审查，若发现考生不符合报考条件，立即取消其复试资格

复试科目相关

2019 年计算机学院/软件学院：

考生的总成绩包括初试成绩和复试成绩，初试成绩和复试成绩各占 50%。

复试成绩中，上机考试成绩占 40%，计算机专业知识面试成绩占总成绩的 50%，

英语面试成绩占总成绩的 10%。复试成绩不及格者，不予录取。

复试成绩不合格者，不予录取。

同等学力考生需加试两门科目：《算法设计与分析》和《软件工程》，闭卷考试

加试成绩不合格者，不予录取



软件学院【全日制】

复试包括专业的综合知识与技能、外语水平及思想政治品德等内容，同时重视本科阶段学习情况和科研能力。复试内容包括上机考试和面试。每位考生的面试复试时间一般不少于 20 分钟，其中外语口试 5 分钟，面试阶段全程录音。同等学力考生需加试两门科目：《数据结构》和《计算机系统基础》，闭卷考试；加试成绩不合格者，不予录取。

机考的考试时间为 150 分钟，闭卷考试，不能携带任何形式的资料。考试内容为程序设计、数据结构和算法，考试环境：Java，C，C++。

软件学院【非全日制】

复试采用面试的形式，主要考察学生对计算机综合知识的掌握和理解能力，应用专业知识分析和解决实际问题的能力等方面。每位考生的复试时间一般不少于 20 分钟，其中英语口语约 5 分钟，进行全程录音。复试全程考生不得使用手机等通讯设备。

计算机学院

复试包括所报考专业的综合知识与技能、外语水平及思想政治品德等内容，同时重视本科阶段学习情况和科研能力。复试内容包括上机考试和面试。每位考生的面试复试时间一般不少于 20 分钟，其中外语口试约 5 分钟，进行全程录音。同等学力考生需加试两门科目：《算法设计与分析》和《软件工程》，闭卷考试；加试成绩不合格者，不予录取。

机考的考试时间为 120 分钟，闭卷考试，不能携带任何形式的资料。考试内容为程序设计、数据结构和算法，考试环境：Visual Studio 2008 C/C++并有 Java 环境。提醒考生：考题中可能会要求从文本文件中读输入，并把结果写入文件。

上机

计算机学院机考的考试时间为 120 分钟，闭卷考试，不能携带任何形式的资料。

软件学院机考的考试时间为 150 分钟，闭卷考试，不能携带任何形式的资料。考试内容为程序设计、数据结构和算法，考试环境：Java, C, C++。

考试内容为程序设计、数据结构和算法

考试环境：Visual Studio 2008 C/C++并有 Java 环境。

提醒考生：考题中可能会要求从文本文件中读输入，并把结果写入文件。

提交的程序必须保证编译正确，编译没通过的程序不评分。

2019 年复旦大学上机题

当时考研最终选择复旦也是考虑到它的复试机试考算法，比较适合我。我本科期间也稍微学过一点算法竞赛。

第一天上午机试遇到了些小状况，就是机房电脑总是出各种毛病，调试了快一个小时才正式开考。我的电脑就出了状况，还是比较影响状态的（甚至手忙脚乱到忘记给志愿信息表勾选“调剂”了）。

机试三道题，也确实不难。主要还是要细心啊。虽然还不知道结果，但我前两题应该还行，各种细节也有考虑到。第三题也不难，但问题在于即便是使用 `long long` 也过不了很多用例。所以最好使用 Java 的大数来写本题。但是很遗憾当我后知后觉地突然意识到 `long long` 不够用的时候已经没时间改 Java 了，才写几行就交卷了。

下午口语面试。我开头的自我介绍也准备的比较充分，虽然临场还是有点忘词。老师也比较和蔼。其中老师还问到我的 “hobby”，我不小心脱口而出 “computer games!” emmmmm 老师我两分钟内能撤回吗。口语问题很宽泛很个性化。比如有大佬被问了乔丹有多高。

第二天一整天面试(99% 的时间在排队)。由于复试人太多，原定的一，二志愿各面试一次，改为只面试一志愿。进房间，一圈的老师，老师超凶_(`_`)_ (明明前面进去的有人出来说老师很和蔼。为什么会变成这样呢?)... 被怼到放弃治疗，我表现真的很差劲。最后临出来前我还是厚着脸皮问老师我机试成绩，“你去问专门的老师。”老师严肃地说到。(无可奉告?!)

复试这两天还是很紧张的，紧张的吃不下饭。

2019 复旦大学计算机学院上机题

1. 相隔天数

题目：输入日期格式：YYYYMMDD，求与 20190205 相隔的天数。

例：

输入：

20190208

输出：

3

2. 最大连续子序列

题目：给定一个数字序列 A_1, A_2, \dots, A_n ，求 $i, j (1 \leq i \leq j \leq n)$ ，使得 $A_i + \dots + A_j$ 最大，

输出这个最大和。

例：

输入：

6

-2 11 -4 13 -5 -2

输出：

20

3. 有向树形态【注意大整数】

题目：求 N 个结点能够组成的二叉树的个数。

例：

输入：3

输出：5

2019 复旦大学工研院上机题

手机按键

题目：

模拟老式手机输入，九宫格布局如下：

```
[ 1 ] [ 2ABC ] [ 3DEF ]  
[ 4GHI ] [ 5JKL ] [ 6MNO ]  
[ 7PQRS ] [ 8TUV ] [ 9WXYZ ]  
[ 0空 ]
```

题目输入为数字或者'-'，其中'-'代表手机输入时等待的时间间隔，数字表示点击某个按键的次数。比如点击两次 2，则输出为 B，四次 2，则输出为 A。

个按键的次数。比如点击两次 2，则输出为 B，四次 2，则输出为 A。

例：

输入：

255

输出：

AK

输入（等待间隔'-'可以无限长，也可以没有）：

2222-----55

输出：

AK

服务器维护

题目：

假设有编号从 1-N 的服务器，首先给出服务器个数，再给出一组服务器状态。

然后给出一组数字，表示修改状态次数，接下来输入为 i, j, x ，意思是使用 x 修改从 i 到 j 的服务器。

最后输出所有服务器状态

例：

输入：

5

1 2 2 3 1

2

1 2 5

2 5 -1

输出：

6 6 1 2 0

计算通讯代价

题目：

给出一个树，计算每个节点到其他节点的通讯代价的总和，假如树为

1———2———3

1

则结点 1, 2, 3 通讯代价分别为：3, 2, 3

例：

输入：

3

1 2

2 3

输出：

3 2 3

输入说明： 3，表示共有 3 个结点，接下来的两行，表示该树节点之间的相连情况。

2018 年复旦大学上机题

总共 3 道。 OJ 提交

1， 求众数。 众数就是一个序列中出现次数最多的数字。 如果不唯一，则输出小的那个值。 给定第一个代表有几个数字。 $1 \leq n \leq 10^5$ 每个数字在 `int` 范围内

样例：

输入， （第一个代表有几个数字）

8

10 3 8 8 3 2 2 2

输出 2

2， 解方程。 给定一个字符串，代表一个 一元一次 方程。 如果有解求解，输出格式 "`x=数字`"，如果解的个数无穷，输出 "`infinite solutions`"。 如果 没有解输出 "`no solution`" 字符串 长度不超过 256 。 样例：

$10x - 2x - 8 = 4x + 7 + x$

输出：

`x=5`

3， 骨牌。 有 $2*n$ 的地板，用 $1*2$ 和 $2*1$ 的骨牌进行铺地板。问共有多少种情况。 结果对 999983 取余 。 $1 \leq n \leq 10000$ 。 样例：

6

输出：

13

我的解法::

1， 第一题本来用 `int num[]` 开一个数组放里面记录次数的。但是后来发现 每个数字在 `int` 范围，开不了那么大，就 `map` 做了

2， 做的时间最长的题目，各位模拟吧。 记录左边的纯数字和，右边的纯数字和。还有 `x` 的两侧的系数和。如果 `x` 的系数不为 0，则有解，如果为 0，再判断左数字和，右数字和是否相同，判断是否解唯一。

注： 这个题我觉得有坑点。 首先就是 256 个字符，估计是大整数的，但是我来不及做，2 个小时 3 道题。 另外就是题目没有说“给定序列一定满足是有意义的”。 也就是说是否可能 没有 `x`，没有等号呢，没有数字呢。 有的就不会成为有效地。 但是 `3=8` 这个要输出 `no solution` 的

2017 年复旦大学上机题

1. 给定一个整数序列，求中位数。
2. 给定一个 9 位数字的 ISBN，求其校验位。ISBN 格式为 2-02-033598，校验位的计算方法如下：从左到右依次将各位数字乘 10, 9, 8,, 2，求出其和 S ，作模运算得 $M = S \bmod 11$ 。若 $11-M$ 在 1 和 9 之间，校验位即为该数字；若 $11-M$ 等于 10，校验位为 X； $11-M$ 等于 11，校验位为 0。输出添加校验位的 ISBN，如 2-02-033598-0。
3. 一个无向图，顶点为 N 个，其中 M 条边已给定，现在要从 K 条备选边中选出若干条，使得整个图连通，且选出的边权值和最小。

2016 年复旦大学上机题

第一题 给定两个字符串，求最大公共子串的长度： 长度小于 1000，两个 `for+string.find` 暴力可解

第二题 给定一个后缀序列，要求求值，只有加减（后缀倒是无所谓）：
水题，后缀直接用栈做就好了，人家复试都是给前缀，要转换的，复旦真是给面子

第三题 是给定一个字符串，求哈夫曼编码的最短长度：

哈夫曼树做，没什么特别要注意的，有好解法，因为你会发现就是求哈夫曼树的非根结点权值之和，或者所有非叶结点之和，想通了这题异常简单，

笨方法建树求高度乘上节点的值也可以，建立指针为叶子指向父结点的树型结构，具体题目要求忘了，记得的可以补充一下

今年的题真的水啊，也就是浙大前三题的难度，今年用了正常的 OJ 无限次提交，两个小时，因为中间在填志愿收材料所以多加了 5 分钟

今年的 oj 可以选择 `c++` 的版本，有 `c++11` 哦，也就是说 `unordered_map/set` 这种神器是可以用的

2015 年复旦大学上机题

今年 3 道题，比起上交浙大的题真的是简单多了，看没人发帖说一说，来凑个热闹哈

1. 给出长方形的长和宽，每次从长方形里撕去最大的正方形，输出最后能得到多少正方形

2. 给出 a, b, c (3 个整数)，判断 a, b 能否通过 $+-*/$ 得到 c ， a, b 可以交换位置，可以输出 YES，不行输出 NO

3. 给出优先队列的实现，实现 4 个操作

- ADD N P: 往队列里加入 id 为 N 的优先级为 P 的任务
- NEXT: 输出下一个最高优先级的任务的 id，如果优先级相同输出 id 小的任务，若队列中没有任务输出 -1
- REMOVE N: 移除 id 为 N 的任务
- COUNT: 输出队列中的任务数量

今年上机用的 oj，可以多次提交，但每道题只能看前两次提交的分数，oj 挺坑的，中间还有会出 bug，结果 lz 也不知道，一直提交，把查看的次数用完了

第一道水题劳资居然超时啊，后来看不到分不敢提交啊，坑

第二道也是水题，第三道直接用 priority_queue 感觉 REMOVE 可能会超时，我用的 set 和 map 做的

往年年复旦大学上机题

2014 年三道题。

1 第一道题目是排序问题。1000 个成绩输出前 30%。最好的做法是用最大堆吧。我用 QSORT 的。算了一下效率差不多。要是数据量的再大估计不行了。

2.第二道题目是二叉树问题。比如节点是 ABCDE 编号是 01234, 给出每个左右子树的编号。求最大叶子间距。我是先建树，然后左右子树深度相加的。

3.第二道是英文题目，为了响应 JYB 的号召。加强英文考核啊。不过是水题。就是给一个字符串比如 ABC 再给一个整数比如 3.输出 AAABBBCCCC 就行了。

第一第三道是黑盒测试。按结果给分。第二道根据算法思想步骤给分。要写注释。

2013 年

Problem1: 字符串匹配

对于主串 M 和模式串 P，找到 P 在 M 中出现的所有子串的的第一个字符在 P 中的位置。P 中第一个字符所在的位置为 0。首行的数字表示有多少组字符串。

[输入及示例]

2

ababababa

ababa

aaa

aa

[输出及示例]

0 2 4

0 1

(相邻位置之间用一个空格隔开)

Problem2:A Famous ICPC Team

Mr. B, Mr. G, Mr. M and their coach Professor S are planning their way for the ACM-ICPC World Finals. Each of the four has a square-shaped suitcase with side length A_i ($1 \leq i \leq 4$) respectively. They want to pack their suitcases into a large square box. The heights of the large box as well as the four suitcases are exactly the same. So they only need to consider the large

box's side length. Of course, you should write a program to output the minimum side length of the large box, so that the four suitcases can be put into the box without overlapping.

[Input]

There are N test cases. The first line is N .

Each test case contains only one line containing 4 integers A_i ($1 \leq i \leq 4$, $1 \leq A_i \leq 1,000,000,000$) indicating the side length of each suitcase.

[Output]

For each test case, display a single line containing the case number and the minimum side length of the large box required.

[Sample Input]

```
2
2 2 2 2
2 2 2 1
```

[Output for Sample Input]

```
Case 1: 4
Case 2: 4
```

[Explanation]

For the first case, all suitcases have size 2×2 . So they can perfectly be packed in a 4×4 large box without wasting any space.

For the second case, three suitcases have size 2×2 and the last one is 1×1 . No matter how you rotate or move the suitcases, the side length of the large box must be at least 4.

Problem3: A Famous Grid

Mr. B has recently discovered the grid named "spiral grid".

Construct the grid like the following figure. (The grid is actually infinite. The figure is only a small part of it.)

Considering traveling in it, you are free to any cell containing a composite number or 1, but traveling to any cell containing a prime number is disallowed. You can travel up, down, left or right, but not diagonally. Write a program to find the length of the shortest path between pairs of nonprime numbers, or report it's impossible.

[Input]

There are N test cases. The first line is N .

Each test case is described by a line of input containing two nonprime integer $1 \leq x, y \leq 10,000$.

[Output]

For each test case, display its case number followed by the length of the shortest path or "impossible" (without quotes) in one line.

[Sample Input]

3

1 4

9 32

10 12

[Output for Sample Input]

Case 1: 1

Case 2: 7

Case 3: impossible

复旦 OJ 题目

数字游戏

发布时间: 2017 年 11 月 12 日 20:06 最后更新: 2017 年 11 月 12 日 20:08 时间限制: 1000ms 内存限制: 512M

描述

一天计算机之神随手写下了一个数列。这个数列的第 N 项是由数字 $1 \sim N$ 依次拼接所得的数。比如第 5 项为 12345，第 11 项为 1234567891011。

现在计算机之神想考考你，这个数列前 N 项有多少数可以被 3 整除。

输入

第一行为一个整数 $T(T \leq 1000)$ ，表示数据组数。

接下来 T 行每行输入一个整数 $N(N \leq 10^9)$ ，表示考虑数列前 N 项。

输出

输出 T 行，每行一个整数代表前 N 项里有多少项能被 3 整除。

样例输入 1

1

5

样例输出 1

3

提示

有：12、123、12345 这三项为 3 的倍数

注意本题数据范围比较大，有 10^9 ，所以请设计一个运算次数小于 10^9 级别的程序来完成这道题。

此后的题目也请注意一下程序的运行效率。

可以认为程序 1 秒能进行 10^8 次操作。

大逃亡

发布时间: 2017 年 11 月 12 日 20:06 最后更新: 2017 年 11 月 12 日 20:08 时间限制: 1000ms 内存限制: 512M

描述

计算机之神正在玩一个惊险刺激的游戏，在游戏中主人公 HS 处在一片极端不稳定的地带。这片地带可以用平面直角坐标系来描述，在时刻 T_i ，坐标点 (X_i, Y_i) 将会发生严重的地面塌陷，连同这个点上下左右四个相邻点 $(X_i+1, Y_i), (X_i-1, Y_i), (X_i, Y_i+1), (X_i, Y_i-1)$ 也将会塌陷。

初始时刻 0 时 HS 处在 $(0,0)$ 位置，他只能平行于坐标轴移动并且移动到相邻坐标点需要 1 个单位时间。现在 HS 要尽快抵达一个位于第一象限或在坐标轴正半轴上的永远不会塌陷的坐标点 (x,y) ，在他逃亡的过程中不能经过已经发生塌陷的坐标点，且任意时刻他所在位置的两个坐标都必须非负。

计算机之神想考考你 HS 能否逃到安全的地方，若能输出最短时间，否则输出 -1。

输入

第一行为一个整数 $T(T \leq 15)$ ，表示数据组数。

接下来 T 组数据每组数据第一行输入一个整数 $M(M \leq 50000)$ ，表示将会发生 M 次塌陷。

接下来 M 行每行三个整数 $X_i, Y_i (0 \leq X_i, Y_i \leq 300), T_i (0 \leq T_i \leq 1000)$ ，表示在时刻 T_i 时位置 (X_i, Y_i) 会发生塌陷。

输出

输出 T 行，每行一个整数代表最少要花多少时间到达安全区域，如果无法到达安全区域则输出 -1。

样例输入 1

```
1
4
0 0 2
2 1 2
1 1 2
0 3 5
```

样例输出 1

```
5
```

提示

一种可行的躲避方案: $(0,0) \rightarrow (0,1) \rightarrow (0,2) \rightarrow (0,3) \rightarrow (0,4) \rightarrow (1,4)$

BubbleSort

发布时间: 2017 年 11 月 12 日 20:06 最后更新: 2017 年 11 月 12 日 20:08 时间限制: 1000ms 内存限制: 512M

描述

最近本能字学园新开始了程序设计课程，Matoi Ryuuko 作为学园的学生也开始练习编程了。她使用 C 语言写出了这样一段冒泡排序代码：

```
for(int i=1;i<=N;++i)
    for(int j=N,t;j>i;--j)
        if(P[j-1] > P[j])
            t=P[j],P[j]=P[j-1],P[j-1]=t;
```

之后她使用这份代码给一个 $1 \sim N$ 的排列 P 进行排序，在排序之后， P 变成了一个升序序列。现在她很好奇对于 $[1, N]$ 中的每个整数 i ，在排序过程（从还没对 P 进行任何操作开始到排序结束）中到达的最右位置-最左位置是多少。

输入

第一行一个整数 T ，代表一共有 $T(1 \leq T \leq 20)$ 组数据。接下来依次给出 T 组数据的内容。每组数据的第一行是一个正整数 $N(1 \leq N \leq 100000)$ ，第二行是 N 个整数，保证是 $1 \sim N$ 的一个排列。

输出

每组数据输出一行 “Case #x: $y_1 y_2 \cdots y_N$ ”，其中 x 是测试数据的编号， y_i =数字 i 的最右位置-最左位置。

样例输入 1

```
2
3
3 1 2
3
1 2 3
```

样例输出 1

```
Case #1: 1 1 2
Case #2: 0 0 0
```

提示

第一组数据的排序过程是这样的：(3, 1, 2) -> (3, 1, 2) -> (1, 3, 2) -> (1, 2, 3)，数字 1, 2, 3 到达的最左和最右位置分别是 1 和 2，2 和 3，1 和 3。

第二组数据一开始就是有序的，所有数字的位置都没有变动。

找质数 - SPJ 测试题

发布时间: 2017 年 11 月 17 日 10:16 最后更新: 2018 年 2 月 9 日 03:20 时间限制: 1000ms 内存限制: 512M SPJ

描述

给一个区间 $[l, r]$, 输出一个在这个区间内的质数。如有多解, 输出任意一个; 如无解, 输出 -1。有 T 组数据。

输入

第一行一个正整数 $T(T \leq 100)$, 表示数据组数。

接下来 T 行, 每行两个正整数 $l, r(1 \leq l \leq r \leq 10^9)$, 表示一组询问。

输出

输出共 T 行, 每行一个数, 表示一组合法答案或者是 -1。

样例输入 1

2

3 3

20 22

样例输出 1

3

-1

提示

SPJ 测试题

A+B Problem

发布时间: 2017 年 11 月 17 日 14:36 最后更新: 2017 年 11 月 22 日 15:52 时间限制: 1000ms 内存限制: 512M

描述

输入两个数 A,B, 输出他们的和

输入

仅一行, 两个正整数 A,B($1 \leq A, B \leq 10^9$)。

输出

仅一行, 一个正整数, 表示 A+B 的结果。

样例输入 1

233 23100

样例输出 1

23333

提示

233+23100=23333

Lucky Number

发布时间: 2017 年 12 月 10 日 22:14 最后更新: 2017 年 12 月 10 日 22:19 时间限制: 1000ms 内存限制: 512M

描述

The 2017 FDU-ICPC contest has started, but Gromah is stuck by a question, so Gromah wants to pick a lucky number and output it directly. Gromah each time will randomly writes an integer k , then find a number that is no more than k and has the largest sum of each digit, which is called the lucky number.

So Gromah wrote T integers k , and he wanted to know what the sum of each digit in the lucky number is for each integer k .

输入

In the first line there is an integer T ($T \leq 1000$), which indicates the number of test cases. In the next T lines there is an integer k ($k \leq 10^9$).

输出

For each test case, print the answer in a line.

样例输入 1

2

19260

817

样例输出 1

36

25

提示

$K = 19260$, the lucky number can be 9999, and the sum of each digit is 36.

$K = 817$, the lucky number can be 799, and the sum of each digit is 25.

Jumping in the City

发布时间: 2017 年 12 月 10 日 22:14 最后更新: 2017 年 12 月 10 日 22:19 时间限制: 1000ms 内存限制: 512M

描述

The god of the computer came to a beautiful city, GM city. The city can be regarded as a rectangle, which is divided into $N \times M$ areas. The top left corner is (1,1) while the lower right corner is (N,M). And the y -th column of the x -th row is denoted as (x, y) . The god has a special ability to jump S areas horizontally or vertically from (x, y) , that is, jumps from (x, y) to $(x, y+S)$, $(x, y-S)$, $(x+S, y)$, $(x-S, y)$, without exceeding the bounds. The god of computer can choose a starting area (x, y) , and then make any number of jumps, but he cannot jump out of the city. Now he wants to select a starting area so that he can get to the largest number of different areas. And he wants to ask you how many of the starting areas allow the largest number of areas to reach.

输入

In the first line there is an integer T ($T \leq 1000$), which indicates the number of test cases. In the next T lines there are three integers N , M and S ($1 \leq N, M, S \leq 10^6$).

输出

Print T lines, on the i -th line print one integer – the answer for i -th test case.

样例输入 1

```
2
2 3 1000000
3 3 2
```

样例输出 1

```
6
4
```

提示

For the first sample case, all the 6 areas can be the answer.

For the second sample case, 4 areas (1,1),(1,3),(3,1),(3,3) can be the answer.

Good Number

发布时间: 2017 年 12 月 10 日 22:14 最后更新: 2017 年 12 月 10 日 22:19 时间限制: 1000ms 内存限制: 512M

描述

FSX often comes up with some strange ideas. Now he gives you a non-negative integer s and defines a positive integer n is a good number of s when it meets the following condition: For every integer from 1 to n , you can choose to add it or minus it. If there's a way to make the final result equal to s , then FSX regards n as a good number of s . Now FSX wants you to find the minimum good number n for each given s .

输入

Input starts with an integer $T(T \leq 10^5)$, denoting the number of test cases.
For each test case, there is only one line describing the given integer $s(0 \leq s \leq 10^9)$.

输出

For each test case output the answer n in one line.

样例输入 1

3
0
1
10

样例输出 1

3
1
4

提示

For the first test case, we can find that $+1+2-3=0$.

For the second test case, we can find that $+1=1$.

For the third test case, we can find that $+1+2+3+4=10$.

Kamui

发布时间: 2017 年 12 月 10 日 22:14 最后更新: 2017 年 12 月 10 日 22:18 时间限制: 4000ms 内存限制: 512M

描述

In order to fight Life Battle-Fibers, little G, the god of computer, created a mysterious organization, but this organization isn't important in this problem. In addition to founding the organization, little G also studies various ways to fight Life Battle-Fibers. The most powerful weapon that he created is a kind of clothing named Kamui. Among all Kamui, Senketsu has the strongest power.

Little G now has the blueprint of Senketsu, but sewing Senketsu is very troublesome, which need to put different kinds of fibers in the correct position. The blueprint can be described by an permutation of integers from 1 to N, with different numbers representing different kinds of fibers. We use 'A' to represent this permutation. If the i-th number in A is j, then the j-th kind of fiber should be put in the i-th position. Little G initially took out n kinds of fibers and placed the i-th kind of fiber in the i-th position. In order to change the fiber sequence to the same as the the blueprint of Senketsu, he can exchange fibers in any two positions any times. If we want to exchange the x-th kind and the y-th kind of fibers, then the exchange needs $W_x + W_y$ units time. Given the array A and W, you need to tell little G the shortest time to complete Senketsu.

输入

In the first line there is an integer T ($1 \leq T \leq 3$), which indicates the number of test cases.

For each test case, the first line is an integer representing N ($1 \leq N \leq 10^6$).

The second line consists of n integers, representing the array W ($1 \leq W_i \leq 10^9$).

The third line consists of n integers, representing the array A. It is guaranteed that A is a permutation of integers from 1 to N.

输出

For each test case, print the answer in a line.

样例输入 1

```
2
4
1 2 1 2
2 1 4 3
6
1 5 5 5 5 5
1 3 4 5 6 2
```

样例输出 1

```
6
36
```

提示

The first case: swap position (1, 2), swap position (3, 4)

The second case: swap position (1,2), swap position (2,6), swap position (5,6), swap position (4,5), swap position (3,4), swap position (1,3).

Palindromic Cut

发布时间: 2017 年 12 月 10 日 22:14 最后更新: 2017 年 12 月 10 日 22:18 时间限制: 1000ms 内存限制: 512M SPI

描述

Phile has a string s of length n consisting of lowercase and uppercase Latin letters and digits.

He wants to rearrange the symbols in s and then cut it into the minimum number of parts so that each part is a palindrome and all parts have the same lengths. A palindrome is a string which reads the same backward as forward, such as [madam] or [racecar].

Please help Phile and determine the minimum number of palindromes of equal lengths to cut s into, if it is allowed to rearrange letters in s before cuttings.

输入

Input starts with an integer $T(T \leq 50)$, denoting the number of test cases.

For each case:

The first line contains an integer $n(1 \leq n \leq 100000)$ — the length of string s .

The second line contains a string s of length n consisting of lowercase and uppercase Latin letters and digits.

输出

For each test case:

Print to the first line an integer k — minimum number of palindromes into which you can cut a given string.

Print to the second line k strings — the palindromes themselves. Separate them by a space. All of them should have the same length.

You are allowed to print palindromes in arbitrary order. And if there are multiple solutions, print any of them.

样例输入 1

3

6

aabaac

8

OrTrT022

2

aA

样例输出 1

2

aba aca

1

02TrrT20

2

a A

Lucky Permutation

发布时间: 2017 年 12 月 10 日 22:14 最后更新: 2017 年 12 月 10 日 22:18 时间限制: 1000ms 内存限制: 512M SPJ

描述

The 2017 FDU-ICPC contest has started, but Gromah is stuck by a question, so Gromah wants to pick a lucky permutation and output it directly. Gromah each time will randomly write down two integers n and k and then find a lucky permutation of $1 \sim n$. From Gromah's perspective, a lucky permutation is such permutation p that for every successive interval of length k , the sum is always a composite number.

Altogether Gromah writes down T pairs of n and k . Now Gromah wants to know whether there exists a lucky permutation for each pair of n and k . If any, print one of them, otherwise print "-1" (without the quotes).

输入

The first line contains a integer T ($T \leq 20$), denoting the number of test cases.

For the following T lines, each line contains two integers n and k ($1 \leq k \leq n \leq 1000$) denoting a test case.

输出

For each test case, print n integers in a line indicating a lucky permutation or print -1 indicating there is no solution.

样例输入 1

2

4 3

4 2

样例输出 1

2 4 3 1

-1

提示

For the first test case, the sum of the interval $[1,3]$ is $2+4+3=9$ and the sum of the interval $[2,4]$ is $4+3+1=8$. 8 and 9 are both composite numbers.

For the second test case, there is no lucky permutation.

Ndb

发布时间: 2017 年 12 月 10 日 22:14 最后更新: 2017 年 12 月 10 日 22:18 时间限制: 1000ms 内存限制: 512M

描述

There are n members in the Ndb($N*d^{***}$ beach) organization, each of whom has its own DTR(DOUTONBORI ROBO). In order to distinguish their DTRs, they decided to mark different numbers on these n DTRs. Ndb founder, little G, believes that only when these n numbers form a geometric sequence, can fully show the charm of Ndb. Due to some technical reasons, the number marked on the DTR can only be an integer in interval $[L,R]$. Now little G wants to know how many different available manners to choose numbers. There are T test cases.

输入

In the first line there is an integer T ($T \leq 10000$), which indicates the number of test cases.

In the next T lines there are three integers n , L and R ($1 \leq n \leq 10^7$, $1 \leq L \leq R \leq 10^7$).

There are only 10 test cases that $n, L, R > 1000$.

输出

For each test case, print the answer in a line.

样例输入 1

2

3 1 10

1 1 10000000

样例输出 1

4

10000000

提示

The first test case: the answer is 4 : {1, 2, 4}, {1, 3, 9}, {2, 4, 8}, {4, 6, 9}.

The second test case: the answer 10000000: {1}, {2}, {3},

Station

发布时间: 2017 年 12 月 10 日 22:15 最后更新: 2017 年 12 月 10 日 22:18 时间限制: 5000ms 内存限制: 512M

描述

Now FSX is in charge of many sea routes. Each route can be represented as a line in the form of $Ax+By=C$ in the two-dimensional plane. Altogether there are n routes and they are neither parallel to each other nor parallel to the coordinate axis. So every two routes have an intersection and in all there are $n(n-1)/2$ intersection points(suppose they are $P_1, P_2, \dots, P_{n(n-1)/2}$). Intersections can be very dangerous so FSX needs to build a station in a point $Q(X_Q, Y_Q)$ to manage these intersections. For convenience, FSX wants the sum of the Manhattan distance between Q and every P_i as small as possible. The definition of Manhattan distance between two points $A(X_a, Y_a)$, $B(X_b, Y_b)$ is $|X_a - X_b| + |Y_a - Y_b|$.

Can you help him find the optimal coordinate of Q ? Note if there are multiple solutions, output the one with the smallest x-coordinate. If there are still multiple solutions, output the one with the smallest y-coordinate.

输入

Input starts with an integer $T(T \leq 30)$, denoting the number of test cases.

For each test case:

The first line contains an integer $n (2 \leq n \leq 40000)$, indicating the number of routes.

For the next n lines, each line contains 3 integers A_i, B_i, C_i , ($1 \leq |A_i|, |B_i| \leq 10^4, 0 \leq |C_i| \leq 10^4$), indicating a route in the form of $A_i x + B_i y = C_i$.

There are only 2 test cases that $n > 1000$.

输出

For each test case, print one line with two floating numbers x, y , both rounded to 5 digits after the decimal point, denoting the coordinate of the station.

样例输入 1

2

3

1 1 1

2 -1 2

-1 2 2

4

1 1 2

1 -1 0

3 -1 -2

1 -3 4

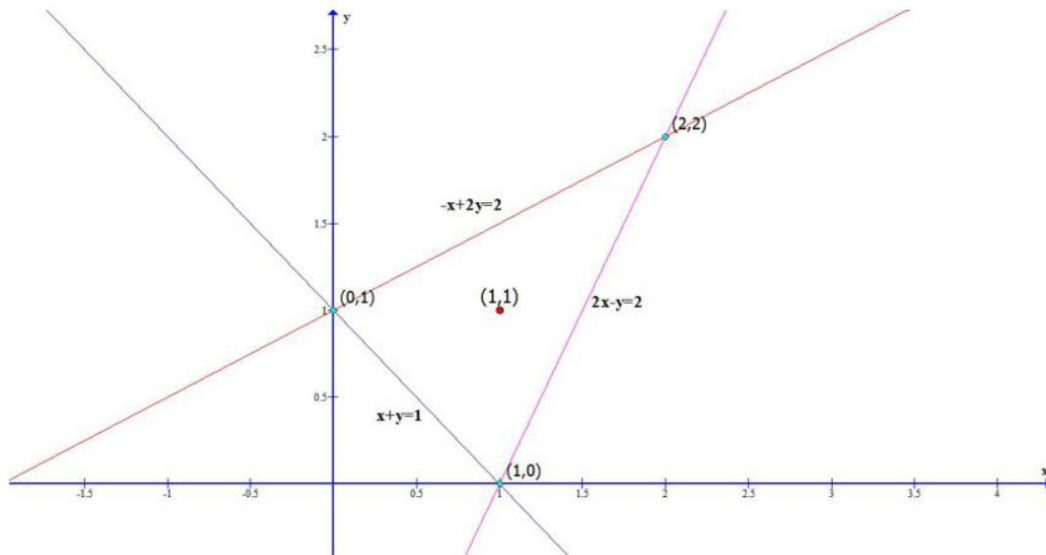
样例输出 1

1.00000 1.00000

-1.00000 -1.00000

提示

For the first test case, we can choose $(1,1)$ as the station since the sum of Manhattan distance between it and the 3 intersections $(0,1),(1,0),(2,2)$ equals 4.



Password

发布时间: 2018 年 12 月 8 日 20:11 最后更新: 2018 年 12 月 8 日 20:12 时间限制: 1000ms 内存限制: 512M

描述

Given a password s with a length of n , we want to figure out how many different characters there are in the password.

输入

The first line of input contains an integer $n(6 \leq n \leq 16)$, representing the length of the password.

The following line contains the string s with the length of n , consisting of only visible characters(with ASCII codes between 33 and 126).

输出

Display the number of different characters.

样例输入 1

14

WelcomeToFDUPC

样例输出 1

12

样例输入 2

11

~!@#\$%^&*()

样例输出 2

11

提示

There are 12 different characters in the first sample, that is, 'W','e','l','c','o','m','t','f','d','u','p' and 'C'.

All characters in the second sample are different.

Coprime

发布时间: 2018 年 12 月 8 日 20:11 最后更新: 2018 年 12 月 8 日 20:12 时间限制: 1000ms 内存限制: 512M

描述

Given an integer n , we suppose that a, b are independent and uniform random integers on the interval $[1, n]$.

Calculate the probability where a and b are coprime.

Note: two integers are said to be coprime if the only positive factor that divides both of them is 1.

输入

A single line with the integer n ($2 \leq n \leq 100$).

输出

Display the most simple fraction (which can be represented in the form of p/q , where p and q are coprime), representing the probability.

样例输入 1

2

样例输出 1

3/4

提示

On 3 cases a and b are coprime that:

$a=1, b=1$;

$a=1, b=2$;

$a=2, b=1$

while the total number of cases is 4.

Nosuri

发布时间: 2018 年 12 月 8 日 20:11 最后更新: 2018 年 12 月 8 日 20:12 时间限制: 1000ms 内存限制: 512M

描述

Nosuri, a thief, wants to unlock a combination lock quickly, which consists of two concentric circle with a center o .

There are n code positions with digits on the outer circle and n input positions on the inner circle.

A code position p corresponds to an input position q if and only if p is on the ray oq . Obviously, an input position's corresponding code position may change if the inner circle is rotated.

The lock will be unlocked if the digits on each input position and its corresponding code position are the same.

Nosuri can do two types of operations:

1. Rotate an input position by $2\pi/10$ clockwise or counter-clockwise. (As we all know, every input position in a combination lock is a regular decagon with 10 digits from 0 to 9.)
2. Rotate the inner circle by $2\pi/n$ clockwise or counter-clockwise.

The number of operations is unlimited. Specially, it can be 0.

Given the digits on each code position and the initial digits on each input position in the inner circle respectively, Nosuri wants to know the minimum number of operations in order to open up the combination clock.

输入

The first line contains an integer $n(1 \leq n \leq 5000)$.

The second line contains n digits. The i -th of them represents the digit on the i -th code position clockwise.

The third line contains n digits. The i -th of them represents the initial digit on the i -th input position clockwise.

Initially, the i -th code position corresponds to the i -th input position.

输出

Output a non-negative integer representing the minimum number of operations.

样例输入 1

3

2 9 5

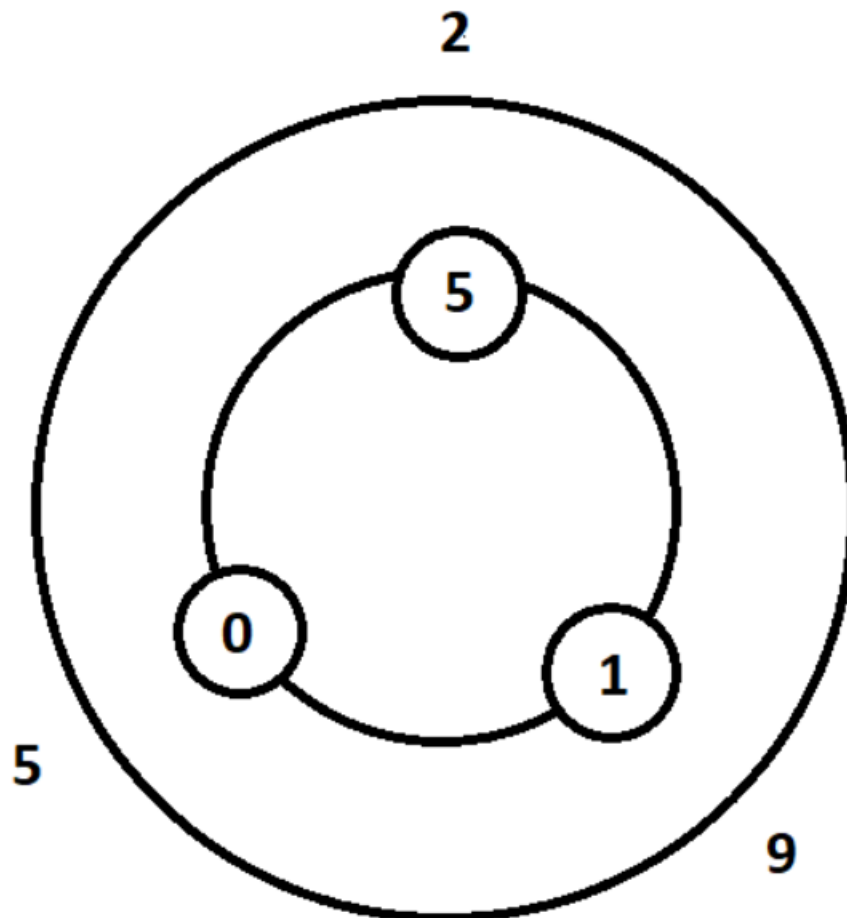
5 1 0

样例输出 1

3

提示

The figure below illustrates the initial state. We first rotate the inner circle counterclockwise, then the input with digit 1, finally the input with digit 0 to unlock in three operations.



Classroom

发布时间: 2018 年 12 月 8 日 20:11 最后更新: 2018 年 12 月 8 日 20:12 时间限制: 1000ms 内存限制: 512M

描述

Given a classroom and n requests, each of which is to borrow the classroom during $[l_i, r_i]$.
How many requests at most can be met at the same time?

输入

Input starts with an integer $n(1 \leq n \leq 105)$, representing the number of requests.

After this are n lines specifying the requests. Each of these lines contains two integers $l_i, r_i(1 \leq l_i \leq r_i \leq 109)$, representing the interval of each these requests, respectively.

输出

Display an single integer representing at most the number of requests met at the same time.

样例输入 1

```
4
1 5
1 5
1 3
4 5
```

样例输出 1

```
2
```

提示

$[1,3], [4,5]$ can be met at the same time.

Range

发布时间: 2018 年 12 月 8 日 20:11 最后更新: 2018 年 12 月 8 日 20:13 时间限制: 1000ms 内存限制: 512M

描述

Given n integers a_1, a_2, \dots, a_n and a const k , how many different pairs (l, r) are there meeting the condition that the range of the subsequence a_l, a_{l+1}, \dots, a_r is strictly greater than k ?

Note: the range of a sequence equals the difference between the maximum and the minimum of the sequence.

输入

The first line of input contains two integers $n, k (1 \leq n \leq 10^5, 1 \leq k \leq 10^9)$.

Following that are n integers $a_i (1 \leq a_i \leq 10^9)$ in the second line.

输出

Display a single integer representing the answer.

样例输入 1

```
5 2
1 2 3 4 5
```

样例输出 1

```
3
```

提示

There are three pairs, $(1, 4), (1, 5), (2, 5)$, which meet the condition.

Finite Demical

发布时间: 2018 年 12 月 8 日 20:11 最后更新: 2018 年 12 月 8 日 20:13 时间限制: 1000ms 内存限制: 512M

描述

Given a decimal fraction in the form of $\frac{1}{q}$, we want to figure out if it can be represented by a finite decimal in k base, where k, q are positive integers and k is greater than 1.

Display "Yes" if it is, or "No" otherwise (without quotes).

输入

A single line with two integers $q, k (1 \leq q, k \leq 10^9)$.

输出

Display "Yes" or "No" (without quotes).

样例输入 1

```
1024 10
```

样例输出 1

```
Yes
```

样例输入 2

```
10 6
```

样例输出 2

```
No
```

提示

In the first sample, $\frac{1}{1024}$ equals to 0.0009765625 in 10-base, which is a finite decimal.

In the second sample, $\frac{1}{10}$ equals to 0.03333... in 6-base, which is an infinite decimal.

Rurutie

发布时间: 2018 年 12 月 8 日 20:11 最后更新: 2018 年 12 月 8 日 20:14 时间限制: 1000ms 内存限制: 512M

描述

Rurutie is now shopping in a bookstore.

There are n books lining up on a bookshelf with ids of $1, 2, 3, \dots, n$ from left to right, where the price of the i th book is a_i , respectively.

Rurutie has w money and wants to spend the money as much as possible due to the fact that more expensive, most interesting.

Rurutie also wants to spend time as little as possible when the condition above has been met.

Getting the books will take Rurutie $r-l$ seconds because she needs to move from the index l to r , where l and r are the smallest and the greatest id of the books she'll buy, respectively.

How long will she spend?

输入

Input starts with two integers $n, w (1 \leq n, w \leq 5000)$.

The second line contains n integers, where the i th integer representing $a_i (1 \leq a_i \leq 5000)$.

输出

Output two integers where the first representing the money Rurutie will spend and the second representing the time she will spend.

Specially, if Rurutie can't buy any books, the second integer should be 0.

样例输入 1

```
4 7
2 1 5 4
```

样例输出 1

```
7 2
```

提示

Rurutie can buy the 1st and the 3rd book with 7 money and 2 seconds.

Kuon

发布时间: 2018 年 12 月 8 日 20:11 最后更新: 2018 年 12 月 8 日 20:17 时间限制: 1500ms 内存限制: 512M

描述

The name of the heroine in this story is Kuon(久遠). Kuon has the similar meaning to "long-term and persistent", so the problem is about the persistent data structure.

For a set with only natural numbers, we define $\text{mex}(S) = \min\{x | x \in \mathbb{N} \wedge x \notin S\}$.

Kuon has a emptyset $Y = \emptyset$ initially.

She'll implement n operations, where the i -th operation is:
1. She'll change Y to the set after the c_i -th operation (Y at that time when the c_i -th operation has been done and the c_i+1 -th operation hasn't been done), or the initial set \emptyset if $c_i=0$.

2. Let $Y = Y \cup \{v_i\}$.

3. Query the value of $\text{mex}(Y)$.

输入

The first line contains an integer representing n ($1 \leq n \leq 5 \cdot 10^6$).

In order to prevent your submissions from TLE during the input phase, c_i ($0 \leq c_i < i$), v_i ($0 \leq v_i \leq 10^9$) are represented in base-64. Here we use `a,b,c,...,z,A,B,C,...,Z,0,1,2,...,9,!,?` to represent the smallest, the second smallest, the third smallest, ..., the 64-th smallest value in base-64, respectively.

The second line contains a string whose length is $4n$. c_i is represented by the $(4i-4)$ -th character to the $(4i-1)$ -th character.

The third line contains a string whose length is $5n$. v_i is represented by the $(5i-5)$ -th character to the $(5i-1)$ -th character.

输出

In order to prevent your submissions from TLE during the output phase, suppose that ansi is the value of $\text{mex}(Y)$ in the i th operation.

You need to output $(\sum_{i=1}^n \text{ans}_i \times 1990214^i) \bmod 1000000009$.

样例输入 1

```
3
```

```
aaaaaaabaaac
```

```
aaaaaaaaabaaac
```

样例输出 1

```
896388709
```

提示

In sample case, $c_1=0, c_2=1, c_3=2, v_1=0, v_2=1, v_3=2$,

and $\text{ans}_1=1, \text{ans}_2=2, \text{ans}_3=3$.

So the output should

be $(1 \times 1990214 + 2 \times 1990214^2 + 3 \times 1990214^3) \bmod 1000000009 = 896388709$.

Also, please be careful of stack overflow.

面试

每位考生的面试复试时间一般不少于 20 分钟，其中外语口试约 5 分钟，**进行全程录音**

数据结构->操作系统->计算机网络->通信原理->微机原理->

软件工程，编译原理，数据库

数据结构

1. 时间复杂度

时间复杂度是指执行算法所需要的计算工作量，因为整个算法的执行时间与基本操作重复执行的次数成正比，所以将算法中基本操作的次数作为算法时间复杂度的度量，一般情况下，按照基本操作次数最多的输入来计算时间复杂度，并且多数情况下我们去最深层循环内的语句所描述的操作作为基本操作。

2. 循环队列的顺序表中，为什么要空一个位置？

这是为了用来区分队空与队满的情况。如果不空一个位置，则判断队空和队满的条件是一样的。

3. 什么是二叉排序树？以及它的原理，算法。（二叉排序树的查找过程）

二叉排序树又称二叉查找树，它或者是一颗空树，或者满足一下性质的二叉树：

- ① 若左子树不空，则左子树上所有结点的值均小于根结点的值；
- ② 若右子树不空，则右子树上所有结点的值均大于根结点的值；
- ③ 左右子树也分别是二叉排序树。

原理步骤：

若根结点的关键字值等于查找的关键字，成功。

否则，若小于根结点的关键字值，递归查左子树。

若大于根结点的关键字值，递归查右子树。

若子树为空，查找不成功。

4. 哈夫曼树

定义：

给定 n 个权值作为 n 个叶子结点，构造一棵二叉树，若带权路径长度达到最小，称这样的二叉树为最优二叉树，也称为哈夫曼树(Huffman tree)。

构造方法：

假设有 n 个权值，则构造出的哈夫曼树有 n 个叶子结点。 n 个权值分别设为 w_1 、 w_2 、...、 w_n ，则哈夫曼树的构造规则为：

- (1) 将 w_1 、 w_2 、...、 w_n 看成是有 n 棵树的森林(每棵树仅有一个结点)；
- (2) 在森林中选出两个根结点的权值最小的树合并，作为一棵新树的左、右子树，且新树的根结点权值为其左、右子树根结点权值之和；
- (3) 从森林中删除选取的两棵树，并将新树加入森林；
- (4) 重复(2)、(3)步，直到森林中只剩一棵树为止，该树即为所求得的哈夫曼树。

特点：

- ① 权值越大的结点，距离根节点越近；
- ② 树中没有度为一的结点。

应用：

哈夫曼编码，减少编码的长度。哈夫曼编码就是长度最短的前缀编码。

5. 什么是哈希冲突？以及如何解决。

散列（哈希）表：

根据关键码值(Key value)而直接进行访问的数据结构。根据给定的关键字来计算出关键字在表中的地址，

以加快查找的速度。

冲突：指的是多个关键字映射同一个地址的情况。

解决办法：

(1) 开放定址法

① 线性探查法（产生堆积问题）；

② 平方探查法（不能探查哈希表上所有的地址，但至少能探查一半的地址）

(2) 链地址法

把所有的同义词用单链表连接起来。

补充（常见的哈希函数构造方法）

直接定址法，数字分析法，平方取中法，除留余数法。

6. 深度优先搜索遍历和广度优先搜索遍历的过程

深度优先搜索遍历

基本思想：首先访问出发点 V ，并将其标记为已访问；然后选取与 V 邻接的未被访问的邻接顶点 W ，访问 W ；再选取与 W 邻接的未被访问的顶点访问，以此类推。当一个顶点所有的邻接顶点都被访问过时，则依次退回最近被访问过的顶点，若该顶点还有其他邻接顶点未被访问，则从这些顶点中去一个顶点进行上述的过程，直至图中所有顶点都被访问过为止。

广度优先搜索遍历

基本思想：首先访问起始顶点 V ，然后选取与 V 邻接的全部顶点 w_1, w_2, \dots, w_n 进行访问，再一次访问与 w_1, w_2, \dots

, w_n 邻接的全部顶点（不包括已访问过的顶点），以此类推，直至所有顶点都被访问过为止。

7. 迪杰斯特拉算法的过程

该算法可以求得某一顶点到其余各顶点的最短路径。

算法思想：设有两个顶点集合 S 和 T ，其中集合 S 中存放的是图中已找到最短路径的顶点，集合 T 中存放的是图中的剩余顶点。

初始状态时，集合 S 中只包含源点 V_0 ，然后不断从集合 T 中选取到顶点 V_0 路径最短的顶点 V_u 并加入集合 S 中。集合 S 每加入一个新的顶点 V_u ，都要修改 V_0 到集合 T 中各个顶点的最短路径的长度值。不断重复这个过程，直至集合 T 中的顶点全部并入到 S 中为止。

8. 链表查找某个元素，平均的时间复杂度是多少？

$O(n)$ 链表是顺序存储，故 $(1+n)/2$ 。

9. 图的存储方式

- ① 邻接矩阵：是图的顺序存储结构，用两个数组分别存储数据元素（顶点）信息和数据元素之间的关系（边/弧）的信息。图的邻接矩阵表示是唯一的，无向图的邻接矩阵是对称的。
- ② 邻接表：是图的链式存储结构，由单链表的表头形成的顶点表和单链表其余结点所形成的边表两部分组成。
- ③ 十字链表：有向图的另一种链式存储结构。
- ④ 邻接多重表：无向图的链式存储结构。

10. 图的深度遍历是否唯一

不一定是唯一。我们可以取图中任一顶点进行深度遍历。

11. 图的相关概念

图：由结点的有穷集合 V 和边的集合 E 组成。

类别：有向图和无向图。

顶点的度：出度和入度。

有向完全图和无向完全图：若有向图有 n 个顶点，则最多有 $n(n-1)$ 条边，则称为有向完全图；

若无向图有 n 个顶点，则最多有 $n(n-1)/2$ 条边，则称为无向完全图。

路径：相邻顶点序列所构成的序列。

简单路径：序列中的顶点和边不重复出现的路径。

回路：路径中第一个顶点和最后一个顶点相同的路径。

连通：无向图中，如果 V_i 到 V_j 有路径，则称这两个顶点连通。如果图中任意两个顶点之间都连通，则称该图为连通图。

有向图中，如果 V_i 到 V_j 有路径，则称这两个顶点连通。如果图中每一对顶点 V_i 和 V_j ，从 V_i 到 V_j 和 V_j 到 V_i 都有路径，则称该图为强连通图。

12. 最小生成树的概念

一个有 n 个结点的连通图的生成树是原图的极小连通子图，且包含原图中的所有 n 个结点，并且有保持图联通的最少的边。如果在最小生成树中添加一条边，必定成一个环。

相关算法：

- ① 普里姆算法
- ② 克鲁斯卡尔算法

N 个结点的最小生成树有几个结点，几条边： n 个结点， $n-1$ 条边。

13. 平衡二叉树

平衡二叉树又称 AVL 树，是一种特殊的二叉排序树，其左右子树都是平衡二叉树，且左右子树的高度差的绝对值不超过 1。

平衡因子：左子树高度减去右子树高度的差。

平衡调整：先找到失去平衡的最小子树，即以距离插入结点最近，且平衡因子绝对值大于 1 的结点最为根结点的子树，分为 LL, LR, RL, RR 四中调节方式。

14. 二叉树的存储

- ① 顺序存储结构：用一个数组来存储一颗二叉树，二叉树中的结点值按照编号依次存入一个一维数组中。适用于完全二叉树，若用于一般的二叉树则会浪费大量存储空间。

Lchild	Data	Rchild
--------	------	--------

- ② 链式存储结构：二叉树中的每一个结点用一个链结点来存放。

15. M 阶 B-树和 M 阶 B+树的主要区别

- ① B+树所有有效数据全在叶子节点，而 B-树所有节点分散在树中，B-树中的关键字不重复。
- ② B+树种有几个关键字就有几个子树，B-树中具有 n 个关键字的节点含有 $(n+1)$ 棵子树。
- ③ B+树有两个指针，根指针和只想最小节点的指针，叶子节点连接成一个不定长的线性链表
- ④ B+树中，每个节点（除根节点外）中的关键字个数 n 的取值范围是 $\lceil m/2 \rceil \leq n \leq m$ ，根节点 n 的取值

- ⑤ 范围是 $2 \leq n \leq m$ 。B-树中, 每个节点 (除根节点外的所有最底层非叶子节点) 中的关键字取值范围是
- ⑥ $\lfloor m/2 \rfloor - 1 \leq n \leq m - 1$, 根节点 n 的取值范围是 $1 \leq n \leq \lfloor m/2 \rfloor$ 。
- ⑦ B+树中的所有非叶子节点仅仅起到索引的作用, 节点中的每个索引项只包含对应子树的最大关键字和指向该子树的指针, 不含有该关键字对应记录的存储地址。而在 B-树中, 每个关键字对应记录的存储地址。

16. 折半查找, 以及其适用范围和时间复杂度

又称二分查找, 基本思路:

在当前的查找区间 $[low...high]$ 中, 首先确定 $mid = (low + high) / 2$, 然后拿关键字与 mid 比较, 若相等则查找成功, 返回该位置, 否则确定新的查找区间, $mid > K, [low...mid - 1]$

$mid < K, [mid + 1...high]$

直至查找自区间长度小于 1 时查找结束。

适用范围: 顺序结构存储并按照关键字大小有序排列。

时间复杂度: $O(\log_2 N)$

17. 完全二叉树

若一棵二叉树至多只有最下面的两层上的结点的度数可以小于 2, 并且最下层上的结点都集中在该层最左边的若干位置上, 则此二叉树成为完全二叉树。

完全二叉树特点:

叶子结点只可能在最大的两层上出现, 对任意结点, 若其右分支下的子孙最大层次为 L , 则其左分支下的子孙的最大层次必为 L 或 $L + 1$;

18. 什么是堆? 有什么作用?

堆是一种数据结构, 可以把堆看成一个完全二叉树, 并且这个完全二叉树满足:

任何一个非叶节点的值都不大于 (或不小于) 其左右子树的结点的值。若父亲大孩子小, 则为大顶堆, 若父亲肖孩子大, 则为小顶堆。

作用: 应用于堆排序。

19. 如何实现循环队列? 有何好处?

如何实现: 把数组弄成一个环, 让 $rear$ 和 $front$ 指针沿着环走, 这样就可以产生循环队列。

好处: 循环队列是顺序队列的改进, 在顺序队列中, 在元素进队的时候, $rear$ 要向后移动, 元素出队的时候, $front$ 也要向后移动, 这样经过一系列的出队和入队操作之后, 两个指针最后会达到数组的末端, 此时虽然队中已经没有元素了, 但是还是不能让元素入队, 即出现了“假溢出”的现象。循环队列就能避免出现这个现象。

20. 深度优先搜索形成的是什么? 森林唯一么?

(森林, 不能说树) (不唯一, 因为邻接表可能不唯一)

21. 满二叉树的结点个数 (n 层)

2 的 n 次方减一 ($2^n - 1$)

22. 二叉查找树查找的时间复杂度以及中序遍历后得到什么样的序列

递增有序序列

23. 什么图可以进行拓扑排序?

有向无环图

24. 顺序队列的特征

队列是一种操作受限的线性表，只允许队尾入队，在队头进行出队。最大的特点是先进先出。

25. 非连通图如何访问每一个结点?**排序总结**

	排序方法	时间复杂度			空间复杂度	稳定性
		平均情况	最坏情况	最好情况		
插入排序	直接插入	$O(n^2)$	$O(n^2)$	$O(n)$	$O(1)$	稳定
	折半插入	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(1)$	稳定
	希尔排序	$O(n \log_2 n)$		$O(n \log_2 n)$	$O(1)$	不稳定
交换排序	冒泡排序	$O(n^2)$	$O(n^2)$	$O(n)$	$O(1)$	稳定
	快速排序	$O(n \log_2 n)$	$O(n^2)$	$O(n \log_2 n)$	$O(\log_2 n)$	不稳定
选择排序	简单选择	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(1)$	不稳定
	堆积排序	$O(n \log_2 n)$	$O(n \log_2 n)$	$O(n \log_2 n)$	$O(1)$	不稳定
其他排序	二路归并	$O(n \log_2 n)$	$O(n \log_2 n)$	$O(n \log_2 n)$	$O(n)$	稳定
	基数排序	$O(d(n+r))$	$O(d(n+r))$	$O(d(n+r))$	$O(r)$	稳定

各类排序的算法详见书本。(需要说出每个算法的基本思想)

查找：相关章节过一遍。

操作系统

1. 进程和程序的区别

- ① 进程是动态的，程序是静止的。进程是程序的执行，程序是有序代码的集合。
- ② 进程是暂时的，程序是永久的。进程是一个状态变化的过程，程序可以长久保存。
- ③ 进程和程序的组成不同：进程包括程序，数据和进程控制块。
- ④ 进程和程序是密切相关的。通过多次执行，一个程序可以对应多个进程；通过调度关系，一个进程可以包括多个程序。
- ⑤ 进程可以创建其他进程，但是程序不能形成新的程序。

2. 进程和线程的区别

- ① 调度：线程是独立调度的基本单位，进程是资源拥有的基本单位。在同一进程中，线程的切换不会引起进程切换。在不同进程中进行线程切换，将会引起进程切换。
- ② 拥有资源：进程是拥有资源的基本单位，而线程不拥有系统资源（除了少量资源，比如栈，程序计数器，寄存器），不过线程可以访问其隶属进程的系统资源。
- ③ 并发性：在引入线程的操作系统中，不仅进程之间可以并发执行，而且同一个进程内的多个线程之间也可以并发执行，能提高系统的吞吐量，系统的并发性也更好。
- ④ 系统开销：在创建进程和撤销进程时，系统都要为之分配或回收资源，所以操作系统为进程付出的系统开销远大于创建线程或撤销线程的开销。
- ⑤ 同步和通信：多线程之间的同步和通信容易实现。

3. 什么是微内核？

微内核操作系统能有效地支持多处理机运行，非常适用于分布式系统环境。

什么是微内核操作系统到现在没有一致公认的定义，但是可以从四个方面对微内核操作系统进行描述：

- ① 足够小的内核：在微内核操作系统中，内核是指精心设计的，能实现现代 OS 最基本核心功能的部分，并非是一个完整的 OS，而只是 OS 中最基本的部分。
- ② 基于 C/S 模式：将操作系统中最基本的部分放入内核中，而把操作系统的绝大部分功能都放于微内核外面的一组服务器中实现。
- ③ 应用“极致与策略分离”原理：在传统 OS 中，讲极致放在 OS 的内核的较低层，把策略放在内核的较高层中。而在微内核 OS 中，通常把机制放在 OS 的微内核中，这样才有可能将内核做得很小。
- ④ 采用面向对象技术。

基本功能：

- ① 进程（线程）管理
- ② 低级存储器管理
- ③ 中断和陷入处理

优点：

- ① 提高了系统的可扩展性
- ② 增强系统的可靠性
- ③ 可移植性
- ④ 提供了对分布式系统的支持
- ⑤ 融入了面向对象技术

4. 什么是 DMA? 什么是中断? 两者的区别。

5. 硬中断和软中断是什么? 区别是什么?

软中断:

- 1、编程异常通常叫做软中断
- 2、软中断是通讯进程之间用来模拟硬中断的一种信号通讯方式。
- 3、中断源发中断请求或软中断信号后,CPU 或接收进程在适当的时机自动进行中断处理或完成软中断信号对应的功能
- 4、软中断是软件实现的中断,也就是程序运行时其他程序对它的中断;而硬中断是硬件实现的中断,是程序运行时设备对它的中断。

硬中断:

- 1、硬中断是由外部事件引起的因此具有随机性和突发性;软中断是执行中断指令产生的,无外部施加中断请求信号,因此中断的发生不是随机的而是由程序安排好的。
- 2、硬中断的中断响应周期,CPU 需要发中断回合信号(NMI 不需要),软中断的中断响应周期,CPU 不需发中断回合信号。
- 3、硬中断的中断号是由中断控制器提供的(NMI 硬中断中断号系统指定为 02H);软中断的中断号由指令直接给出,无需使用中断控制器。
- 4、硬中断是可屏蔽的(NMI 硬中断不可屏蔽),软中断不可屏蔽。

区别:

- 1、软中断发生的时间是由程序控制的,而硬中断发生的时间是随机的
- 2、软中断是由程序调用发生的,而硬中断是由外设引发的
- 3、硬件中断处理程序要确保它能快速地完成它的任务,这样程序执行时才不会等待较长时间。

6. 页面置换算法有哪些? 什么是 LRU?

- ① 最佳置换算法 (OPT): 在预知一个进程的页面号引用串的情况下,每次都淘汰以后不再使用的页以后最迟再被使用的页面。该算法不能实现,只能作为一个标准来衡量其他置换算法的优劣。
- ② 先进先出算法 (FIFO): 每次总是淘汰最先进入内存的页面,也就是将在内存中驻留时间最长的页面淘汰。(可能会产生 Belady 异常,缺页次数随着分配的物理块的增加而增加)。
- ③ 最近最少使用算法 (LRU): 选择最近最少未被使用的页面淘汰,其思想是用以前的页面引用情况来预测将来会出现的页面引用情况。利用了局部性原理。
- ④ 时钟置换算法 (CLOCK): 是 LRU 和 FIFO 的折中,具体方法略。
- ⑤ 工作集算法
- ⑥ 工作集时钟算法
- ⑦ 第二次机会算法
- ⑧ 最近未使用 (NRU)

7. 操作系统中的磁盘调度算法

磁盘调度算法目的：使磁盘的平均寻道时间最少。

调度算法	算法思想	优点	缺点
先来先服务算法 FCFS	按照进程请求访问磁盘的先后顺序进行调度。	简单，公平。	未对寻道进行优化，平均寻道时间较长，仅适用于磁盘请求较少的场合。
最短寻道时间优先算法 SSTF	选择与当前磁头所在磁道距离最近的请求作为下一次服务的对象。	较 FCFS 有较好的寻道性能以及较少的寻道时间。	会导致饥饿现象
扫描（电梯调度）算法 SCAN	在磁头当前移动方向上选择与当前磁头所在磁道距离最近的请求最为下一次服务的对象。	具有较好的寻道性能，而且防止了饥饿现象。	存在一个请求刚好被错过，需要等待很久的情形。
循环扫描算法 CSCAN	规定磁头单向移动，如自里向外移动，当磁头移动到最外的磁道时立即返回到最里磁道，如此循环进行扫描。	兼顾较好的寻道性能，防止饥饿现象，同时解决了一个请求等待时间过长的的问题。	可能出现磁臂长时间停留，处理不懂的情况（磁臂黏着）
N-Step-SCAN 算法，对 SCAN 算法的优化。	将磁盘请求队列分成若干个长度为 N 的子队列，磁盘调度将按照 FCFS 依次处理这些子队列，而每处理一个队列时又是按照 SCAN 算法，对一个队列处理后再处理其他队列，将新请求队列放入新队列。	无磁臂黏着。	
FSCAN 算法，对 SCAN 算法的优化。	将请求队列分成两个子队列，将新出现请求磁盘 IO 的进程放入另一个子队列。	无磁臂黏着。	

8. 操作系统中的信号量

信号量是一个确定的二元组 (s, q) ，其中 s 是一个具有非负初值的整型变量， q 是一个初始状态为空的队列。整型变量 s 表示系统中某类资源的数目，当其值大于 0 时，表示系统中当前可用资源的数目；当其值小于 0 时，其绝对值表示系统中因请求该类资源而被阻塞的进程数目。

信号量分类：

- ① 整型信号量：所谓整型信号量就是一个用于表示资源个数的整型量
- ② 记录型信号量（资源信号量）：就是用了一个结构体实现，里面包含了表示资源个数的整型量和一个等待队列。

信号量的应用：

- ① 实现进程同步
- ② 实现进程互斥

9. PV 操作

信号量的值除了初值外，仅能由这 PV 原语加以改变。P、V 操作以原语形式实现，保证了对信号量进行操作过程中不会被打断或阻塞。P 操作相当于申请资源，V 操作相当于释放资源。P 操作和 V 操作必定成对出现，但未必在同一个进程中。

```

Struct semaphore{
    Int count;
    queueType queue;
};

Wait (semaphore s)      // P
{
    s.count --;
    if(s.count<0)
    {    阻塞该进程;
        将该进程插入等待序列 s.queue;
    }
}

signal (semaphore s)    // V
{
    s.count ++;
    if(s.count<=0)
    {    从等待队列 s.queue 取出第一个进程 p;
        将 p 插入就绪队列;
    }
}

```

10. 什么是操作系统？

操作系统（Operating System，简称 OS）是管理和控制计算机硬件与软件资源的计算机程序，是直接运行在“裸机”上的最基本的系统软件，任何其他软件都必须在操作系统的支持下才能运行。操作系统是用户和计算机的接口，同时也是计算机硬件和其他软件的接口。操作系统的功能包括管理计算机系统的硬件、软件及数据资源，控制程序运行，改善人机界面，为其它应用软件提供支持等，使计算机系统所有资源最大限度地发挥作用，提供了各种形式的用户界面，使用户有一个好的工作环境，为其它软件的开发提供必要的服务和相应的接口。

11. 操作系统的组成

12. 操作系统中用到了哪些数据结构中的数据结构？请举例说明

13. 简述操作系统中系统调用过程

系统调用提供了用户程序和操作系统之间的接口，应用程序通过系统调用实现其余 OS 的通信，并取得它的服务。系统调用不仅可供所有的应用程序使用，而且也可供 OS 本身的其它部分，如命令处理程序。

系统调用的处理步骤（三步）：

首先，将处理机状态由用户态转为系统态；然后由硬件和内核程序进行系统调用的一般性处理，即首先保护被中断进程的 CPU 环境，将处理机状态字 PSW、程序计数器 PC、系统调用号、用户栈指针以及通用寄存器内容等压入堆栈；再然后将用户定义的参数传送到指定的地址保存起来。

其次，分析系统调用类型，转入相应的系统调用处理子程序。（通过查找系统调用入口表，找到相应处理子程序的入口地址转而去执行它。）

最后，在系统调用处理子程序执行完后，应恢复被中断的进程设置新进程的 CPU 现场，然后返回被中断进程或新进程，继续往下执行。

14. 虚拟存储器，以及相关算法。

基于局部性原理，应用程序在运行之前，仅将那些当前要运行的少数页面或段先装入内存便可运行，其余部分暂时留在盘上。程序运行时，如果它要访问的页已调入内存，便可继续执行下去；但如果程序要访问的页或段尚未调入内存（即缺页），此时程序应利用请求调入功能将它们调入内存，以使程序能继续执行下去。如果此时内存已满，无法装入新的页或段，则需要利用页面置换功能，将内存中暂不使用的页面或段调至盘上，腾出空间用于页面调入内存，是程序继续执行下去。这样，就实现了大的用户程序能在较小的内存空间里运行，也可以在内存中同时装入更多的进程使它们并发运行。从用户角度出发，该系统的内存容量比实际内存容量大很多，故成这样的存储器为虚拟存储器。

相关算法：

页面置换算法

- ① 最佳置换算法 (OPT)：在预知一个进程的页面号引用串的情况下，每次都淘汰以后不再使用的页以后最迟再被使用的页面。该算法不能实现，只能作为一个标准来衡量其他置换算法的优劣。
- ② 先进先出算法 (FIFO)：每次总是淘汰最先进入内存的页面，也就是将在内存中驻留时间最长的页面淘汰。（可能会产生 Belady 异常，缺页次数随着分配的物理块的增加而增加）。
- ③ 最近最少使用算法 (LRU)：选择最近最少未被使用的页面淘汰，其思想是用以前的页面引用情况来预测将来会出现的页面引用情况。利用了局部性原理。
- ④ 时钟置换算法 (CLOCK)：是 LRU 和 FIFO 的折中，具体方法略。
- ⑤ 工作集算法
- ⑥ 工作集时钟算法
- ⑦ 第二次机会算法
- ⑧ 最近未使用 (NRU)

15. 存储器管理应具有的功能

存储管理的主要任务是为多道程序的运行提供良好的环境，方便用户使用存储器，提高存储器的利用率以及从逻辑上扩充存储器，故应具有以下功能：

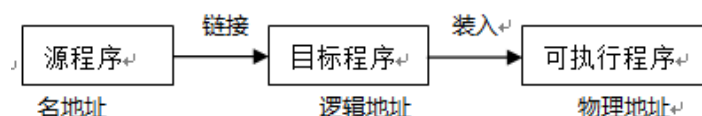
- ① 内存的分配和回收：实施内存的分配，回收系统或用户释放的内存空间。
- ② 地址变换：提供地址变换功能，将逻辑地址转换成物理地址。
- ③ 扩充内存：借助于虚拟存储技术或其他自动覆盖技术，为用户提供比内存空间大的地址空间，从逻辑上扩充内存。
- ④ 存储保护：保证进入内存的各道作业都在自己的存储空间内运行，互不干扰。

16. 什么是 TLB?

TLB 的作用是在处理器访问内存数据的时候做地址转换。TLB 的全称是 Translation Lookaside Buffer，可以翻译做旁路缓冲，是一个具有并行查询能力的特殊高速缓冲寄存器。TLB 中存放了一些页表文件，文件中记录了虚拟地址和物理地址的映射关系。当应用程序访问一个虚拟地址的时候，会从 TLB 中查询出对应的物理地址，然后访问物理地址。TLB 通常是一个分层结构，使用与 Cache 类似的原理。处理器使用一定的算法把最常用的页表放在最先访问的层次。

17. 程序的装入方式有哪些?

补充：应用程序从用户编写的源文件到内存中执行的进程大致分为三个阶段，经过编译程序将源代码便以为若干个目标模块，在通过链接程序将编译好的目标模块以及所需的库函数链接到一起，形成完整的装入模块，最后通过装入程序将这些装入模块装入内存并执行。（编译，链接，装入）



装入方式：

- ① 绝对装入：在编译时就知道程序将要驻留在内存的物理地址，编译程序产生含有物理地址的目标代码，不适合多道程序设计。
- ② 可重定位装入：根据内存当前情况，将装入模块装入到内存的适当位置，地址变换通常在装入时一次完成，之后不再改变，也称静态重定位。当操作系统为程序分配一个以某地址为起始地址的连续主存区域后，重定位时将程序中指令或操作数的逻辑地址加上这个起始地址就得到了物理地址。
- ③ 动态运行装入：允许程序运行时在内存中移动位置，把装入模块装入到内存后的所有地址都是相对地址，在程序执行过程中每当访问到相应指令或数据时，才将要访问的程序或数据的相对地址转换为物理地址。动态重定位的实现要依靠硬件地址变换机构。

18. 程序的链接方式有哪些?

- ① 静态链接：在程序运行之前，先把各个目标模块及所需库链接为一个完整的可执行程序，以后不再拆开。
- ② 装入时动态链接：将应用程序编译后所得到的一组目标模块在装入内存时采用边装入边链接的链接方式。
- ③ 运行时动态链接：知道程序运行过程中需要一些模块时，才对这些模块进行链接。

19. 交换技术，覆盖技术，以及两者的区别。

覆盖技术：把一个大的程序划分为一系列覆盖，每个覆盖是一个相对独立的程序单位，把程序执行时并不要求同时装入内存的覆盖组成一组，成为覆盖段，这个覆盖段分配到同一个存储区域，这个存储区域成为覆盖区，它与覆盖段一一对应。覆盖段的大小由覆盖段中最大的覆盖来确定。（为了解决内存容量太小的问题，打破了必须将一个程序全部信息装入内存后才能运行的限制）

交换技术：把暂时不用的某个程序及数据部分从内存移到外存中去，以便腾出必要的内存空间；或者把指定的程序或数据从外存读到相应的内存中，并将控制权交给他，让其在系统上运行的一种内存扩充技术。处理器的中级调度就是采用交换技术。

区别：

- ① 与覆盖技术相比，交换技术不要求程序员给出的程序段之间的覆盖结构；

- ② 交换技术主要在进程和作业之间进行，覆盖技术主要在同一个进程或作业中进行；
- ③ 覆盖技术只能覆盖于覆盖程序段无关的程序段，交换进程由换出和换入两个过程组成。

20. 内存连续分配管理方式有哪些？

- ① 单一连续分配（静态分配）
- ② 固定分区分配（分区大小可以不等，但事先必须确定，运行时不能改变）
- ③ 动态分区分配

P131 详细

21. 动态分区分配的算法有哪些？

- ① 首次适应算法 First Fit
- ② 循环首次适应算法 Next Fit
- ③ 最佳适应算法 Best Fit
- ④ 最差适应算法 Worst Fit

22. 什么叫拼接技术？

在分区管理方式下，系统运行一段时间后，内存中会出现相当一部分的碎片，拼接技术是解决碎片问题的方法。

即将存储器中所有已分配分区移动到主存的一端，使本来分散的多个小空闲区连成一个大的空闲区，这种通过移动把多个分散的小分区拼接成一个大分区的方法即为拼接技术。

23. 什么叫原子操作？

24. 内部碎片和外部碎片

内部碎片：分配给作业的存储空间中未被利用的部分。

外部碎片：系统中无法利用的小存储块，比如通过动态内存分配技术从空闲内存区上分配内存后剩下的那部分内存块。

25. 常用的存储保护方法

- (1) 界限寄存器

上下界寄存器方法

基址、限长寄存器方法

- (2) 存储保护键：给每个存储块分配一个单独的存储键，它相当于一把锁。

26. 连续分区分配和非连续分区分配的比较

27. 什么是页表？有什么作用。

为了便于在内存中找到进程的每个页面所对应的物理块，系统为每个进程建立一张页面映射表。

页表由页表项组成，页表项有页号和块号组成，根据页表项就可以找到每个页号对于物理内存中物理块的块号。

28. 什么是段寄存器？

29. 进程线程树图

进程树是一个形象化的比喻，比如一个进程启动了一个程序，而启动的这个进程就是原来那个进程的子进程，依此形成的一种树形的结构，我们可以在进程管理器选择结束进程树，就可以结束其子进程和派生的子进程。

30. 作业和进程的区别

一个进程是一个程序对某个数据集的执行过程，是分配资源的基本单位。作业是用户需要计算机完成的某项任务，是要求计算机所做工作的集合。一个作业的完成要经过作业提交、作业收容、作业执行和作业完成 4 个阶段。而进程是对已提交完毕的程序所执行过程的描述，是资源分配的基本单位。

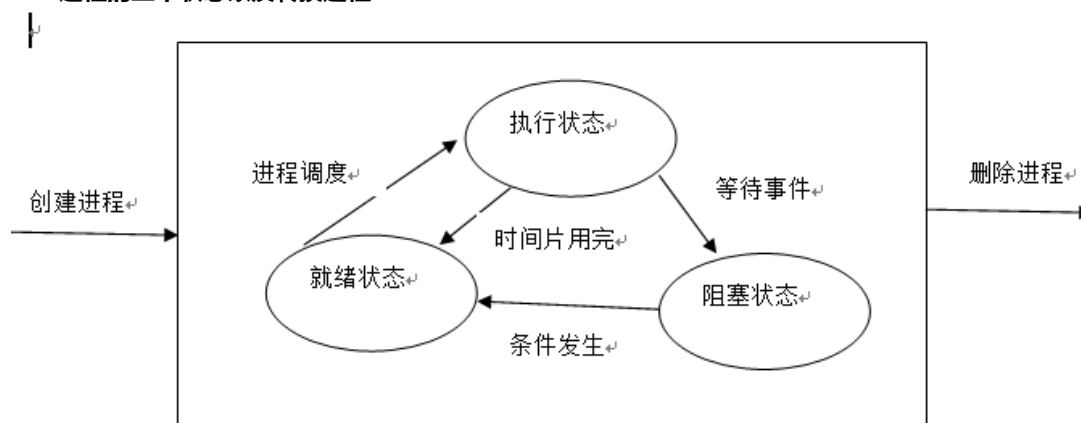
其主要区别如下。

(1) 作业是用户向计算机提交任务的任务实体。在用户向计算机提交作业后，系统将它放入外存中的作业等待队列中等待执行。而进程则是完成用户任务的执行实体，是向系统申请分配资源的基本单位。任一进程，只要它被创建，总有相应的部分存在于内存中。

(2) 一个作业可由多个进程组成，且必须至少由一个进程组成，反过来则不成立。

(3) 作业的概念主要用在批处理系统中，像 UNIX 这样的分时系统中就没有作业的概念。而进程的概念则用在几乎所有的多道程序系统中进程是操作系统进行资源分配的单位。在 Windows 下，进程又被细化为线程，也就是一个进程下有多个能独立运行的更小的单位。

31. 进程的三个状态以及转换过程



32. 进程调度算法有哪些？（再详细了解其中的算法思想）

- ① 先来先服务调度 FCFS
- ② 短作业优先调度 SJF
- ③ 优先级调度 Priority

- ④ 时间片轮转调度 RR
- ⑤ 高响应比优先调度
- ⑥ 多级队列调度
- ⑦ 多级反馈队列调度

33. 死锁

死锁：是指两个或两个以上的进程在执行过程中，因争夺资源而造成的一种互相等待的现象，若无外力作用，它们都将无法推进下去。此时称系统处于死锁状态或系统产生了死锁，这些永远在互相等待的进程称为死锁进程。

死锁原因：

- ① 系统资源不足
- ② 进程推进顺序不当

产生死锁的必要条件：

- ① 互斥条件：指进程对所分配到的资源进行排它性使用，即在一段时间内某资源只由一个进程占用。
- ② 请求和保持条件：指进程已经保持至少一个资源，但又提出了新的资源请求，而该资源已被其它进程占有，此时请求进程阻塞，但又对自己已获得的其它资源保持不放。
- ③ 不剥夺条件：指进程已获得的资源，在未使用完之前，不能被剥夺，只能在使用完时由自己释放。
- ④ 环路等待条件：指在发生死锁时，必然存在一个进程——资源的环形链，即进程集合{P0, P1, P2, ..., Pn}中的 P0 正在等待一个 P1 占用的资源；P1 正在等待 P2 占用的资源，……，Pn 正在等待已被 P0 占用的资源。

处理死锁的基本方法：

- ① 预防死锁：这是一种较简单和直观的事先预防的方法。方法是通过设置某些限制条件，去破坏产生死锁的四个必要条件中的一个或者几个，来预防发生死锁。预防死锁是一种较易实现的方法，已被广泛使用。但是由于所施加的限制条件往往太严格，可能会导致系统资源利用率和系统吞吐量降低。
- ② 避免死锁：该方法同样是属于事先预防的策略，但它并不须事先采取各种限制措施去破坏产生死锁的四个必要条件，而是在资源的动态分配过程中，用某种方法去防止系统进入不安全状态，从而避免发生死锁。
- ③ 检测死锁：这种方法并不须事先采取任何限制性措施，也不必检查系统是否已经进入不安全区，此方法允许系统在运行过程中发生死锁。但可通过系统所设置的检测机构，及时地检测出死锁的发生，并精确地确定与死锁有关的进程和资源，然后采取适当措施，从系统中将已发生的死锁清除掉。
- ④ 解除死锁：这是与检测死锁相配套的一种措施。当检测到系统中已发生死锁时，须将进程从死锁状态中解脱出来。常用的实施方法是撤销或挂起一些进程，以便回收一些资源，再将资源分配给已处于阻塞状态的进程，使之转为就绪状态，以继续运行。

34. 什么是饥饿？与死锁有什么差别？

等待时间给进程推进和响应带来明显影响时成为进程饥饿。

饥饿并不代表系统一点死锁，但至少有一个程序的执行被无限期地推迟。

差别：

- ① 进入饥饿的进程可以只有一个，但是死锁必须大于等于两个；
- ② 出于饥饿状态的进程可以是一个就绪进程，但是死锁状态的进程必定是阻塞进程。

35. 分段和分页的区别

36. 银行家算法

主要思想是避免系统进入不安全状态，在每次进行资源分配时，它首先检查系统是否有足够的资源满足要求，如果有，则先试行分配，并对分配后的新状态进行安全性检查。如果新状态安全，则正式分配上述资源，否则拒绝分配上述资源。这样就保证系统始终处于安全状态，从而避免死锁现象的发生。

37. RAID 磁盘阵列

独立磁盘冗余阵列（RAID, redundant array of independent disks, redundant array of inexpensive disks）是把相同的数据存储在多个硬盘的不同的地方（因此，冗余地）的方法。通过把数据放在多个硬盘上，输入输出操作能以平衡的方式交叠，改良性能。因为多个硬盘增加了平均故障间隔时间（MTBF），储存冗余数据也增加了容错。

38. 控制管理模块是写在哪个文件里的？那个文件叫做什么文件？

1. 连接 2 个局域网需要什么 在那一层

2. TCP 与 UDP 的连接区别及适用情况

3. 路由和交换的区别

4. 七层网络结构

物理层

数据链路层 (PPP、HDLC、CSMA/CD)

网络层(IP、ARP (IP→MAC) /RARP(MAC→IP)、ICMP)

传输层 (TCP、UDP)

会话层

表示层

应用层 (telnet 23、FTP 20<数据>+21<控制>、SMTP 161、DNS、SNMP、DHCP、HTTP 80)

TCP/IP 是四层 (网络接口层、网际层、运输层、应用层)

5. 时分复用的时隙

6. IPV4 和 IPV6 的位数

IPV4 是 32 位; IPV6 是 128 位

7. 单工、半双工、全双工

单工: 又称为单向通信, 即只能有一个方向的通信而没有反方向的交互。例: 无线电广播, 电视广播

半双工: 又称为双向交替通信, 即通信的双方都可以发送信息, 但不能双方同时发送 (当然也就不能同时接受)。

全双工: 又称为双向同时通信, 即通信的双方可以同时发送和接受信息。

备注: 单工只要一条信道, 而半双工和全双工都需要两条信道 (每个方向各一条)。

9. 网络通信过程

10. 简述一下停等协议

由于 IP 层是不可靠的, 因此 TCP 需要采取措施使得传输层之间的通信变得可靠。停止等待协议就是保证可靠传输, 以流量控制为目的的一个协议。其工作原理简单的说就是每发送一个分组就停止发送, 等待对方的确认, 在收到确认后再发送下一个分组, 如果接受方不返回应答, 则发送方必须一直等待。

细节描述: ①如果发送方如果一段时间仍没有收到确认, 就认为刚才发送的分组丢失了, 因而重传前面发送过的分组。

②如果接收方的确认分组丢失或者因其他原因, 收到了重传分组, 则: 丢弃这个重传分组, 并且向发送方发送确认。

11. 应用层有什么协议, 举出两个协议的作用

DNS(域名系统), 将网址解析成 IP 地址。

DHCP (动态主机设置协议), 是一个局域网的网络协议, 使用 UDP 协议工作, 主要有两个用途: 给内部网络或网络服务供应商自动分配 IP 地址, 给用户或者内部网络管理员作为对所有计算机作中央管理的手段。

FTP(文件传输协议), FTP 客户机可以给服务器发出命令来下载文件, 上载文件, 创建或改变服务器上的目录。

TELNET(远程终端协议), 能将用户的击键传到远程主机, 同时也能将远程主机的输出通过 TCP 连接返回到用户屏幕。

12. 数据链路层的作用

数据链路层在物理层提供的服务的基础上向网络层提供服务，将原始的、有差错的物理线路改进成逻辑上无差错的数据链路，从而向网络层提供高质量的服务。功能主要有：①链路管理；②封装成帧；③透明传输；④差错检测

13. 路由协议有哪些

域内：

①RIP（路由信息协议）。距离矢量协议；16跳；收敛速度慢；适用小规模网络；仅和邻居交换信息，且交换的是整个路由表，按固定时间周期性交换；实现简单，开销小；传输层使用UDP；RIP报文限制大小为512字节。

②OSPF（开放最短路由协议）。链路状态协议；基于IP；向自治域中所有路由器发送信息（防洪法），且发送的是链路状态表（LSA），触发更新；更新过程收敛快；根据代价选择最佳路由；支持多路径负载均衡。

域间：

BGP（外部网关协议）是不同AS之间的路由器之间交换路由信息的协议。

14. 频分复用如何避免各路信号间的干扰

15. 简述计算机网络中各层作用 这问题回答之后 老师一直追着网络层问了好多 问得我发懵了。。

16. 列举数据链路层的协议。。2个即可

局域网：

①CSMA/CD（带冲突检测的载波侦听多路访问协议），每一个站在发送数据之前要先检测一下总线上是否有其他计算机在发送数据，如果有，则暂时不发送数据，以免发生冲突；如果没有，则发送。并且计算机在发送数据的同时边检测信道上是否有冲突发生。如果有，则采用截断二进制指数类型退避算法来等待一段随机时间后再次重发。总的来说，可概括为“先听后发，边听边发，冲突停发，随机重发”。

广域网：

①PPP（点对点协议），面向字节；不需要的功能：纠错（PPP协议只负责检错）、流量控制（由TCP负责）、序号（PPP协议是不可靠传输协议，故不需要对帧进行编号）、多点线路（PPP协议是点对点的通信方式）、半双工或单工（PPP只支持全双工链路）。

②HDLC（高级数据链路控制协议），面向比特；标记字段（01111110），地址字段（全1是广播，全0为无效）；控制字段（根据前两位取值可以将HDLC帧划分为三类（无奸细）：信息帧，监督帧，无编号帧），信息字段，帧检验序列FCS（CRC循环冗余码）

17. 网络各层的设备分别是什么

物理层：集线器，中继器（均不能隔离冲突域和广播域）

数据链路层：交换机（转发/学习机制、存储转发、隔离冲突域不能隔离广播域），网桥

网络层：路由器（隔离冲突域和广播域）

18. 什么是滑动窗口协议

滑动窗口协议，是TCP使用的一种流量控制方法。该协议允许发送方在停止并等待确认前可以连续发送多个分组。由于发送方不必每发一个分组就停下来等待确认，因此该协议可以加速数据的传输。

19. PPP 协议

20. 网络按地理范围分

个域网、局域网、城域网、广域网

21. 保护频带 就是插入一些 空白的频段

22. 一个网络安全有哪些方面，还有个 p2p 协议

23. DNS DHCP DNS 倒是记得是让网址转换 IP 地址的 映射 IP 地址 动态主机。。协议

DHCP 是动态分配 ip 吗

24. 流量控制在哪些层实现

传输层 (TCP)、数据链路层 (ARQ)

25. 频分复用 时分复用 波分复用 码分复用

频分复用：给每个信号分配唯一的载波频率并通过单一媒体来传输多个独立信号的方法。

时分复用：把多个信号复用到单个硬件传输信道，它允许每个信号在一个很短的时间使用信道，接着再让下一个信号使用。

波分复用：就是光的频分复用。用一根光纤同时传输多个频率很接近的光载波信号。

码分复用：码分复用是用一组包含互相正交的码字的码组携带多路信号。每一个用户可以在同样的时间使用同样的频带进行通信。由于各用户使用经过特殊挑选的不同码型，各用户之间不会造成干扰，因此这种系统发送的信号有很强的抗干扰能力。

26. CSMA/CD 协议 如果两端同时发送信息会出现什么情况，为什么？

两端都检测到冲突，均停止发送数据，等待一个随机时间再重发。

27. 电路交换，分组交换

见高分笔记 P27

29. 简述下 CSMA/CD 协议的实现原理

30. 描述网络某一层的原理

31. 说一下路由器的原理

传统地，路由器工作于 OSI 七层协议中的第三层，其主要任务是接收来自一个网络接口的数据包，根据其中所含的目的地址，决定转发到下一个目的地址。因此，路由器首先去掉数据包的二层头，取出目的 IP 地址，在转发路由表中查找它对应的下一跳地址，若找到，就在数据包的帧格前添加下一个 MAC 地址，同时 IP 数据包头的 TTL (Time To Live) 域也减一，并重新计算校验和。当数据包被送到输出端口时，它需要按顺序等待，以便被传送到输出链路上。

计算机网络

1. 网络里时延和带宽的概念

时延是指一个报文或分组从一个网络的一端传送到另一个端所需要的时间。它包括了发送时延，传播时延，处理时延，排队时延。(时延=发送时延+传播时延+处理时延+排队时延)一般，发送时延与传播时延是我们主要考虑的。对于报文长度较大的情况，发送时延是主要矛盾；报文长度较小的情况，传播时延是主要矛盾。

带宽又叫频宽，是指在固定的时间可传输的资料数量，亦即在传输管道中可以传递数据的能力。在数字设备中，频宽通常以 bps 表示，即每秒可传输之位数。在模拟设备中，频宽通常以每秒传送周期或赫兹 (Hz)来表示 (指通过的频率的宽度)。

2. 网络拥塞

网络中存在太多的数据包导致数据包被延迟或丢失，从而降低了整个网络的传输性能，这种情况叫做拥塞。如果产生网络拥塞，则网络的性能明显下降，整个网络的吞吐量将随着输入负荷的增大而下降。

拥塞控制：可分为闭环控制和开环控制。开环控制是在设计网络时事先将有关拥塞的因素考虑到，力求网络在工作时不产生拥塞。闭环拥塞是基于反馈环路的概念。

拥塞控制的 4 种算法：

- ① 慢开始算法
- ② 拥塞避免算法
- ③ 快重传算法
- ④ 快恢复算法

3. CSMA/CD 的原理 (载波侦听多路访问/冲突检测方法)

工作原理：发送数据前，先侦听信道是否空闲，若空闲，则立即发送数据，在发送数据时，边发送边继续侦听，若侦听到冲突，则立即停止发送数据，等待一段随机时间，再重新尝试。**(先听后发，边发边听，冲突停发，随机延迟后重发。)**

4. 三网指哪三网？

电信网，广播电视网，互联网。

5. 组成网络协议的三个要素

- ① 语义：对构成协议元素的含义的解释；
- ② 语法：数据域控制信息的结构和格式；
- ③ 同步：规定事件的执行顺序。

6. 电路交换，报文交换，分组交换之间的区别以及联系

	电路交换	报文交换	分组交换
建立被双方独占物理通路 (建立连接)	需要	不要求	不要求
数据交换单位	比特流	报文	分组
传输方式	比特流直接到达	存储转发	存储转发

每个分组沿着规定路径	是		不是
分组按序到达	是		不是
优点	通信时延小; 实时性强; 适用范围广; 控制简单; 避免冲突。	无需建立连接; 动态分配路线; 可靠性高; 线路利用率高; 可提供多目标服务。	加速传输; 简化存储管理; 减少出错几率和重发数据
缺点	建立连接时间长; 信道利用率低; 缺乏统一标准; 灵活性差。	由于存储转发方式导致转发时延; 对报文大小没有控制,需要较大的存储缓存空间。	存在传输时延; 工作量大,可能出现分组丢失等情况。

7. 分组交换的优点和缺点

与报文交换相比较, 分组交换的优点和缺点如下。

优点:

- ① **加速传输**: 因为分组交换是逐个传输, 所以可以使后一个分组的存储操作与前一个分组的转发操作并行, 这种流水线式传输方式减少了报文的传输时间, 此外, 传输一个分组所需的缓冲区比传输一个报文所需的缓冲区小得多, 这样因缓冲区不足而等待发送的几率及等待的时间也必然减少。
- ② **简化了存储管理**: 分组的长度固定, 相应的缓冲区大小也固定。
- ③ **减少出错几率和重发数据量**: 因为分组较短, 其出错几率必然减少, 所以每次重发的数据量也就大大减少, 这样不仅提高了可靠性, 也减少了传输时延。

缺点:

- ① **存在传输时延**: 尽管与报文交换相比的传输时延较短, 但是与电路交换相比仍然存在存储转发时延, 而且对其结点交换机的处理能力有更高的要求。
- ② **当分组交换采用数据报服务时**, 可能出现失序、丢失或重复分组, 分组到达的目的地的结点时, 要对分组按编号进行排序工作, 工作量较大。

8. 流量控制在哪些层实现?

流量控制就是要控制发送方数据传输的速率, 使接收方来得及接收。

数据链路层: 相邻结点的流量控制,

传输层: 端到端的流量控制, 利用滑动窗口机制在 TCP 连接上实现流量控制。

9. CDMA 及原理

码分多路复用, 既共享信道的频率, 又共享时间, 是一种真正的动态复用技术。

原理:

10. 二层交换机和三层交换机的区别

二层交换技术是发展比较成熟, 二层交换机属数据链路层设备, 可以识别数据包中的 MAC 地址信息, 根据 MAC 地址进行转发, 并将这些 MAC 地址与对应的端口记录在自己内部的

一个地址表中。

三层交换机就是具有部分路由器功能的交换机

三层交换技术就是二层交换技术 + 三层转发技术。传统交换技术是在 OSI 网络标准模型第二层——数据链路层进行操作的，而三层交换技术是在网络模型中的第三层实现了数据包的高速转发，既可实现网络路由功能，又可根据不同网络状况做到最优网络性能。

11. 频分复用如何避免频带间的干扰，保护频带？

12. 停止等待协议

13. IPv4 地址缺乏的解决办法以及 IPv4 的替代方案以及 IPv4 和 IPv6 如何相互通信？

14. 传统的搜索引擎基本原理，基于内容的搜索？原理和实现？

15. 什么是非对称加密？什么是数据安全的特征？

16. 网络安全有哪些方面

17. 网络的分类

分类	作用范围	使用范围	拓扑结构
	局域网，城域网，广域网，互联网	公用网，专用网	星状，网状，总线，树形，令牌环

18. 有关于信道划分的问题

多路复用：当传输介质的带宽超过了传输单个信号所需的带宽时，通过在一条介质上同时携带多个传输信号的方法来提高传输系统的利用率。

多路复用技术能把多个信号组合在一条物理信道上进行传输，使多个计算机或终端设备共享信道资源，提高信道利用率。

频分多路复用 FDM	所有用户在同样的时间占用不同的带宽资源 为了防止子信道之间的干扰，相邻信道之间要加入保护频带
时分多路复用 TDM, STDM 统计时分复用	所有用户在不用的时间占用相同的频带资源； 统计时分复用是一种动态的时间分配，又称异步时分复用。
波分多路复用 DWDM	每个用户在同一时间使用同样的频带进行通信
码分多路复用	既共享信道频率，又共享时间。

CDMA

信道共享技术：

信道共享技术又称为多点接入（multiple access）技术，包括随机接入和受控。

从层次上讲，信道共享是由数据链路层的媒体接入控制 MAC 子层来完成的。总的来说，在计算机网络中使用的信道共享技术可以分为三种，即随机接入、受控接入和信道复用。

- ① 随机接入，特点是所有的用户都可以根据自己的意愿随机地向信道上发送信息。当两个或两个以上的用户都在共享的信道上发送信息的时候，就产生了冲突(collision)，它导致用户的发送失败。随机接入技术主要就是研究解决冲突的网络协议。随机接入实际上就是争用接入，争用胜利者可以暂时占用共享信道来发送信息。随机接入的特点是：站点可随时发送数据，争用信道，易冲突，但能够灵活适应站点数目及其通信量的变化。典型的随机接入技术有 ALOHA、CSMA、CSMA/CD。将会在后面章节中详细介绍。
- ② 受控接入，特点是各个用户不能随意接入信道而必须服从一定的控制。又可分为集中式控制和分散式控制。

集中式控制的主要方法是轮询技术，又分为轮叫轮询和传递轮询，轮叫轮询主机按顺序逐个询问各站是否有数据，传递轮询主机先向某个子站发送轮询信息，若该站完成传输或无数据传输，则向其临站发轮询，所有的站依次处理完后，控制又回到主机。

分散式控制的主要方法有令牌技术，最典型的应用有令牌环网，其原理是网上的各个主机地位平等，没有专门负责信道分配的主机，在环状的网络上有一个特殊的帧，称为令牌，令牌在环网上不断循环传递，只有获得的主机才有权发送数据。

- ③ 信道复用，指多个用户通过复用器(multiplexer)和分用器(demultiplexer)来共享信道，信道复用主要用于将多个低速信号组合为一个混合的高速信号后，在高速信道上传输。其特点是需要附加设备，并集中控制，其接入方法是顺序扫描各个端口，或使用中断技术。

19. 曾经问到过的协议

ICMP、PPP、DNS、DHCP、

20.

21. 计算机网络各层设备及工作原理总结

层	设备	工作原理	其他
物理层	集线器, 中继器	中继器: 作用就是讲已经衰竭得不完整的信号经过整理, 重新产生出完整的信号再继续传送。(放大器放大的是模拟信号)	集线器不能分割冲突域
		集线器 Hub: 即多端口的中继器, 假设一台 8 个接口的集线器连接 8 台计算机, 计算机 1 发送信息给计算机 8, 计算机 1 的网卡会将信息通过双绞线传送到集线器中, 集线器将信息进行广播, 其他 7 个端口的计算机接收到广播时, 会对信息进行检查, 如果是自己的则接受否则不予理睬。	
数据链路层	交换机, 网桥	网桥: 网桥至少有两个端口, 每个端口与一个网段连接, 网桥每从一个端口接收到一个帧, 就先暂存在缓存中, 若该帧未出现差错, 且欲发往的目的地 MAC 地址属于另一个网段 (同一网段无需转发, 应该丢弃), 则通过查找转发表, 从相应端口发出。	
		交换机: 实质上是一个多端口网桥, 每个端口都直接与主机或集线器相连, 并且一般都工作在全双工方式。当主机需要通信时, 交换机能同时连通许多对的端口, 使每一对相互通信的主机都能像独占通信媒体一样, 进行无冲突地传输数据, 通信完成后断开连接。	
网络层	路由器	路由器的主要作用是转发数据包, 通过查询其的路由信息表, 将每一个 IP 数据包由一个端口转发到另一个端口。	

设备名称	隔离冲突域	隔离广播域
集线器	✗	✗
中继器	✗	✗
交换机	✓	✗
网桥	✓	✗
路由器	✓	✓

22. 计算机网络协议总结 (按层总结: 什么层, 协议, 作用, 特点等)

层	传输单位	任务	功能	设备	协议
物理层	比特	透明地传输比特流	为数据端设备提供传送数据通路	集线器/中继器	
数据链路层	帧	将网络层传下来的 IP 数据报组成帧	① 链路的建立, 拆除, 分离; ② 帧定界和帧同步 ③ 差错检测	交换机/网桥	PPP、HD ARQ
网络层	数据包	① 将传输层传下来的报文段封装成分组 ② 选择适当的路由, 是传输层传下来的分组能够交付到目标主机	为传输层提供服务; 组包和拆包; 路由选择; 拥塞控制。	路由器	ICMP ARP RARP IP IGMP
传输层	报文段 TCP,UDP	负责主机两个进程之间的通信	为端到端提供可靠的传输服务; 为端到端连接提供流量控制、差错控制、服务质量等管理服务。		TCP UDP

会话层	应用层		提供系统与用户的接口	文件传输； 访问和管理； 电子邮件服务。		FTP、SMTP、POP3、HTTP、DHCP、RDP、BGP
-----	-----	--	------------	----------------------------	--	---------------------------------

计算机组成，微机原理，通信原理

1. RAM 和 ROM 的原理和区别（在基本存储单元上存在本质区别）

RAM:随机通道存储器（random access memory）。一般用来存放各种现场的输入输出数据、中间计算结果、与外存交换的信息以及作为堆栈。一般分为双极型 RAM 和 MOS RAM（又分为静态 SRAM、动态 DRAM）。

双极型 RAM 以晶体管的触发器作为基本存储电路，所以管子多，速度快、功耗大、集成度较低、成本高，一般用于高速计算机或者 cache。

MOS RAM 一般分为静态 SRAM 和动态 DRAM。静态 RAM 用由六管构成的触发器作为基本电路，集成度适中，功耗适中，速度较快，不需要刷新电路（为了维持数据，动态 RAM 中需要一个刷新电路，在短周期内对所有基本存储单元进行充电，维持电平状态，来达到数据的保存）。

动态 DRAM 基本存储电路用单管线路组成（电平靠电容存储，如一般计算机中的内存），集成度高，功耗低，成本比 SRAM 低，由于靠电容的电荷保持电平，所以需要刷新电路（典型要求每隔 1ms 刷新一次）。

总的说 RAM 使用晶体管打开或关闭通往位于交点处的电容器来访问电路。它有如下特性：A、易挥发 B、速度快 C、成本高，即不易制作大容量的 RAM。所以这类存储器一般用在计算机设备的内存部分，而且往往容量不会很大。

ROM: read-only memory，只读存储器。一般用来存放固定的程序以及存放各种常数、函数表。一般分为掩模 ROM、可编程的 ROM（PROM）和可重复擦除的 ROM 三种（按出现时间顺序排列）。

掩模 ROM: 由厂家固化一次性程序，不可擦写。原理是：由晶体管构成，‘0’：存储单元通过晶体管导通连地，上电后该单元视为‘0’电平。‘1’：与‘0’逻辑相反，存储单元没有晶体管接地。出现时间最早。

PROM: 可编程 rom，本身不带有程序，只允许用户进行一次擦写操作，出现时间较晚。

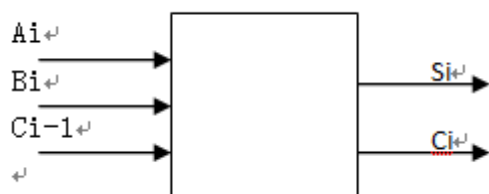
可擦写可编程 rom：有紫外线擦除（EPROM，70 年代初）、电擦除（EEPROM）以及 FLASH（闪存）。他们都是可以多次重复擦写的 ROM。

EPROM 原理是通过击穿绝缘层向浮空栅注入电子进行写操作；利用紫外线使浮空栅电子泄露进行擦操作。

EEPROM: 原理与 EPROM 相似，在绝缘层间加入隧道二极管，于是可以通过电场作用，使浮空栅带上电子或消去。

FLASH：（太高端了，我也不懂 0.0。）

2. 一位全加器的真值表以及逻辑表达式



上图是一位全加器示意图：Ai、Bi 是两个二进制加数，Ci-1 是上一级加法器的进位，Si 是本级一位加法输出，Ci 是加法进位。真值表如下：

$$S_i = A_i \oplus B_i \oplus C_{i-1}$$

$$C_i = (A_i \oplus B_i) C_{i-1} + A_i B_i \quad (\oplus \text{表示异或逻辑})$$

3. 什么是芯片组

芯片组 (Chipset) 是构成主板电路的核心。一定意义上讲，它决定了主板的级别和档次。它就是"南桥"和"北桥"的统称，就是把以前复杂的电路和元件最大限度地集成在几颗芯片内的芯片组。芯片组是整个身体的神经，芯片组几乎决定了这块主板的功能，进而影响到整个电脑系统性能的发挥，芯片组是主板的灵魂。芯片组性能的优劣，决定了主板性能的好坏与级别的高低。这是因为目前 CPU 的型号与种类繁多、功能特点不一，如果芯片组不能与 CPU 良好地协同工作，将严重地影响计算机的整体性能甚至不能正常工作。

北桥芯片：提供对 CPU 类型和主频的支持、系统高速缓存的支持、主板的系统总线频率、内存管理（内存类型、容量和性能）、显卡插槽规格，ISA/PCI/AGP 插槽、ECC 纠错等支持；

南桥芯片：提供了对 I/O 的支持，提供对 KBC（键盘控制器）、RTC（实时时钟控制器）、USB（通用串行总线）、Ultra DMA/33(66)EIDE 数据传输方式和 ACPI（高级能源管理）等的支持，以及决定扩展槽的种类与数量、扩展接口的类型和数量（如 USB2.0/1.1，IEEE1394，串口，并口，笔记本的 VGA 输出接口）等；

4. 触发器相关的问题

三类：电位触发、主从触发、边沿触发
等等。。。

5. 中断的软件实行过程

中断源产生中断请求，CPU 一般要先屏蔽该中断源的中断，防止错误的中断嵌套->CPU 对现场进行保存，存储断点程序地址并将当前数据压入栈中->PC 指向对应的中断入口，转入执行中断向量指向的中断服务程序->完成中断响应后，恢复现场，程序回到断点处，将栈中的数据重新读出->重新开中断。

6. 查询传输和中断传输，解释比较

中断传输：中断式传送是指处理器进行的各种处理。执行主程序时，输入设备准备好数据，或者输出设备已把数据输出，向处理器发出中断申请，使处理器中止当前程序的执行，转向中断服务程序，读取输入设备的数据，或者把数据送往输出设备。

无条件传送和查询传输：程序控制传送分为无条件传送（同步传送）和程序查询传送（条件传送、异步传送）。**无条件传送**只对固定的外设（如开关、继电器、7 段显示器等）在规定的时间内用 IN 或 OUT 指令进行信息的传输，实质是用程序定时同步传送数据。**条件查询**的条件是在执行输入或输出指令前，要先查询接口中状态寄存器的状态。输入时，由它只是要输入的数据是否已准备就绪；而输出时，由它只是输出设备是否空闲，由此条件决定执行输入或输出。

可以看出，从 CPU 的角度看，中断的响应相对查询来说是被动的、实时的，如网络的通讯请求，键盘、鼠标的操作等等。查询则一般用于主动的、可控的场合，通常在需要输入/输出数据时，对外设的状态进行查询，以防冲突。而很多外设的速度是很慢的（如打印机），CPU 就会长时间处于等待的状态，无法进行其他操作，这就降低了 CPU 的效率，所以检测到这类设备忙碌的状态时，我们会用设置中断的方法避免这种等待，提高 CPU 的使用率。

中断是难以或者无法预测的，查询的操作是用户在程序中制定的，所以一般是可以控制的。

7. 过程调用具体执行了哪些操作

8. PC 机的端口是同步的还是异步的？什么是异步？

USB 好像是同步的。

PC 机的串口是异步的，即 UART，采用 RS-232 标准。

异步通信：它用一个起始位表示字符的开始，用停止位表示字符的结束来构成一帧。它的传输帧是由起始位、内容和停止位构成的。每个字符可能还包含有奇偶校验位。波特率一般在 50-9600 baud。数据信号传输速率单位是 字符/秒 或 bps。

同步通信：为了提高速度，去掉异步通信中使用的起始位和停止位，而在数据块开始前使用一段同步字符指示通信开始，同步通信对整块数据进行传输，速度较快，通常为几十到几百 Kbaud。

9. 控制单元设计·分为组合逻辑和微程序·两者区别·优缺点·

电路的输出仅由该时刻电路的输入决定的电路，称为组合逻辑电路。

上面那句是概念性的东西，具体在数字电路组合逻辑部分。

郎艳峰上的 EDA 技术用的 FPGA 就是组合逻辑控制的一种，不过那个太低端了。。

程序的特点是：为了达到一定目的，程序按照一定的逻辑关系一条一条地执行，程序执行的快慢取决于机器周期的快慢、CPU 的性能以及软件算法的优劣等。

1、**速度上**：组合逻辑电路由用户制定好逻辑关系后，输入信号经过电路的响应，直接得出结果，很大程度上是纯硬件的响应，因此在时刻间隙内可以完成可能需要通过多条程序才能完成的算法，不难看出，组合逻辑电路设计的控制单元比程序控制的单元具有更快的处理速度，而且系统越复杂，组合逻辑在速度上的优势越明显。

2、**复杂度上**：组合逻辑电路是由设计者按照自己的设想将众多逻辑器件组合而成的，对于较为复杂的系统，设计一个逻辑电路往往需要庞大的工作量，检验和仿真等都是费时费力的工程。在这一点上，程控的方法显得方便的多，程序结构的有条不紊是它在设计复杂度上占优势的主要因素。一个无比复杂的逻辑结构可能只需要设计者为 CPU 设计一些精炼的算法就可以实现，在对处理速度没有十分苛刻的要求情况下，通过程序设计会大大地减轻设计者的工作量。

3、**成本**：一般来说，大型可编程逻辑器件的成本要比一些较为高端的处理器或单片机高得多，而且工作量一般比较大，花费的工比较多。

总的来说，工程设计上，要结合实际应用的要求和成本，设计者要尽量选择全方位的设计方案。

10. 关于 IO 接口的，微机原理中的内容，要仔细看下。

接口：接口是位于主机和外设之间的一种缓冲电路，包括了硬件电路和软件控制，现在接口通常为可编程的大规模集成电路芯片。

端口：接口电路通常有三种寄存器：数据输入输出缓冲寄存器、控制寄存器、状态寄存器。这些寄存器称为（数据、控制、状态）端口。对 I/O 设备的访问，就是通过对端口访问实现的，每个端口赋予一个唯一的地址码，称为端口地址。

端口的编址：独立编址和统一编址。统一编址是将端口当做存储器单元，与内存统一编址，内存和端口的操作指令是统一的，所以指令功能强，但是端口占用了存储器的地址空间，使存储器容量减少。

独立编址也称为 I/O 映射编址，即端口和内存分开各自独立地拥有自己的地址空间，端口不占用存储器空间。

80x86 采用独立编址。

I/O 端口地址译码：1、用逻辑门电路进行译码 2、用译码器进行译码 3、比较器译码 4、PROM（可编程 rom）和 GAL（通用逻辑阵列）。

11. 中断（概念，补充：中断可不可以被打断，有哪些情况）

指当出现需要时，CPU 暂时停止当前程序的执行转而执行处理新情况的程序和执行过程。即在程序运行过程中，系统出现了一个必须由 CPU 立即处理的情况，此时，CPU 暂时中止程序的执行转而处理这个新的情况的过程就叫做中断。

在响应中断时，CPU 执行两个连续的中断响应周期，每个周期都输出中断响应信号/INTA，第一个周期 CPU 使 0~15 位的地址总线(8086 有 20 根地址总线)浮空，第二个周期被响应的外设向数据总线输送一个字节的 interrupt 向量号，而后 CPU 读取中断向量号，转向中断服务程序，响应中断。

有些不可屏蔽中断是可以打断其它中断的。如掉电、复位等。

23. 简述 RAM、ROM、PROM、EPROM、EEPROM 的区别（第一题说过了...）

24. 什么是指令，时钟，总线周期，有什么关系

我们把要求计算机执行的各种操作用命令的形式写下来，就是指令。

历年复试细则

2019 年复旦大学计算机科学技术学院硕士研究生招生复试实施细则

为做好我院 2019 年度招收公开招考硕士生工作，根据复旦大学下发的相关工作文件精神，结合学院实际情况，拟定本方案。

一、复试名单的确定

由学院研究生招生工作领导小组依据学校公布的考生参加复试的基本要求、当年的招生计划和考生的初试成绩择优拟定复试名单。复试考生名单在我院网站上公布。

为避免高分低能和有利于优胜劣汰，我院复试采取差额形式，差额比例一般不低于 120%。

二、成立复试小组

学院高度重视复试工作，加强领导，精心组织。按学科方向成立复试小组，各学科方向选派责任心强、教学经验丰富、学术及外语水平较高的人员参与复试工作，复试小组成员一般不少于 5 人。复试小组成员应认真贯彻“公平公正，客观准确”原则，严格把好复试的质量关。

三、招生监察

学院党委选派合适人员承担招生监察工作，安排纪检人员或指定专人列席面试，对复试工作实施全程监督和重点时段监督。

四、报考资格审查

按照《复旦大学 2019 年硕士考生报考资格审查要求》，考生应携带以下材料参加复试：

(1) 考生的准考证和有效身份证件原件。

(2) 应届生的学生证原件。

(3) 往届生的学历证书原件和教育部学历证书电子注册备案表（或学历认证报告），学位证书原件（报考对学位有要求专业的考生）。

(4) 持在境外获得学历/学位的考生，须提交教育部留学服务中心出具的认证报告（境外学校在读本科生，不能以应届生身份报考）。

(5) 对于同等学力考生，须查验授课学校教务部门提供的 8 门相关专业本科课程考试成绩单原件。

(6) 对于在读研究生，须查验培养单位出具的同意报考证明；考生还需于拟录取前提供注销原学籍证明。

(7) 对于在 2019 年 9 月 1 日前可取得国家承认本科毕业证书的自学考试和网络教育本科生，须查验颁发毕业证书的省级高等教育自学考试办公室或网络教育高校出具的相关证明。

(8) 对于少数民族骨干计划考生，须查验是否具有《报考 2019 年少数民族高层次骨干人才计划硕士研究生考生登记表》。没有《登记表》的，不能列入少数民族骨干计划。

(9) 对于“退役大学生士兵”专项计划考生，须查验是否具有《入伍批准书》和《退出现役证》。

2. 在资格审查时，考生须携带上述材料原件和复印件各一套，复印件由院系留存。

经审查，若发现考生不符合报考条件，立即取消其复试资格，初试成绩无效。

五、复试的内容、时间地点及方式

复试包括所报考专业的综合知识与技能、外语水平及思想政治品德等内容，同时重视本科阶段学习情况和科研能力。复试内容包括上机考试和面试。每位考生的面试复试时间一般不少于 20 分钟，其中外语口试约 5 分钟，进行全程录音。同等学力

力考生需加试两门科目：《算法设计与分析》和《软件工程》，闭卷考试；加试成绩不合格者，不予录取。

机考的考试时间为 120 分钟，闭卷考试，不能携带任何形式的资料。考试内容为程序设计、数据结构和算法，考试环境：Visual Studio 2008 C/C++ 并有 Java 环境。提醒考生：考题中可能会要求从文本文件中读输入，并把结果写入文件。

复试时间:3 月 18 日和 19 日，地点：张衡路 825 号复旦大学张江校区，具体安排在研究生报考服务系统中通知。

六、复试评分标准

考生的总成绩包括初试成绩和复试成绩，初试成绩和复试成绩各占 50%。

复试成绩中，上机考试成绩占 40%，计算机专业知识面试成绩占总成绩的 50%，英语面试成绩占总成绩的 10%。复试成绩不及格者，不予录取。

七、填报学科方向志愿和分组面试

计算机专业知识面试分为以下 5 个学科方向，即：计算机科学与技术、网络空间安全、软件工程、大数据试验场、医学信息学。

其中，计算机科学与技术一级学科包括：计算机系统结构、计算机软件与理论和计算机应用技术专业，以及（专业学位）计算机技术的智能媒体计算、智能信息处理和智能网络与系统 3 个研究方向。网络空间安全一级学科包括：网络空间安全和（专业学位）计算机技术的网络空间安全基础理论、密码学与数据安全和网络与系统安全 3 个研究方向。软件工程一级学科包括：（专业学位）计算机技术的软件工程与系统软件、数据科学与数据工程和人机交互与协同计算 3 个研究方向。大数据试验场为（专业学位）计算机技术的大数据技术与应用研究方向，医学信息学为（专业学位）计算机技术的临床医学信息处理研究方向。

学院在复试前向考生公布各学科方向的招生名额，复试前考生在学院规定的时间内填写第一志愿和第二志愿，选择是否愿意调剂，并分别参加报考学科方向的面试。

复试小组成员现场独立打分。复试后，各面试小组将面试的成绩上报学院，由学院计算出考生总成绩。学院根据总成绩、考生填报的志愿次序，确定拟录取名单。

八、考生咨询方式和监督与申诉渠道

考生咨询方式：计算机科学技术学院教务办公室，联系电话：021-51355555-

14，邮箱：cs_jiaoxue@fudan.edu.cn。

监督与申诉渠道：计算机科学技术学院党委办公室，联系电话：021-51355555-

28，邮箱：cs_dangban@fudan.edu.cn。

本细则自公布之日起实施，由计算机科学技术学院负责解释。

复旦大学计算机科学技术学院

2019 年 3 月 11 日

2019 年复旦大学软件学院全日制硕士研究生招生复试实施细则

为做好我院 2019 年度招收公开招考硕士生工作, 根据复旦大学下发的相关工作文件精神, 结合学院实际情况, 拟定本方案。

一、复试名单的确定

由学院研究生招生工作领导小组依据学校公布的考生参加复试的基本要求、当年的招生计划和考生的初试成绩择优拟定复试名单, 复试采取差额形式, 差额复试比例一般不少于 120%。复试考生名单在我院网站上公布。

二、成立复试小组

学院高度重视复试工作, 加强领导, 精心组织。按学科方向成立复试小组, 各学科方向选派责任心强、教学经验丰富、学术及外语水平较高的人员参与复试工作, 复试小组成员一般不少于 5 人。复试小组成员应认真贯彻“公平公正, 客观准确”原则, 严格把好复试的质量关。

复试小组负责确定复试的内容, 负责复试命题和评定复试成绩。

三、招生监察

学院党委选派合适人员承担招生监察工作, 安排纪检人员或指定专人列席面试, 对复试工作实施全程监督和重点时段监督。

四、报考资格审查

按照《复旦大学 2019 年硕士考生报考资格审查要求》, 考生应携带以下材料参加复试:

- (1) 考生的准考证和有效身份证件原件。
- (2) 应届生的学生证原件。
- (3) 往届生的学历证书原件和教育部学历证书电子注册备案表(或学历认证报告), 学位证书原件(报考对学位有要求专业的考生)。
- (4) 持在境外获得学历/学位的考生, 须提交教育部留学服务中心出具的认证报告(境外学校在读本科生, 不能以应届生身份报考)。
- (5) 对于同等学力考生, 须查验授课学校教务部门提供的 8 门相关专业本科课程考试成绩单原件。
- (6) 对于在读研究生, 须查验培养单位出具的同意报考证明; 考生还需于拟录取前提供注销原学籍证明。
- (7) 对于在 2019 年 9 月 1 日前可取得国家承认本科毕业证书的自学考试和网络教育本科生, 须查验颁发毕业证书的省级高等教育自学考试办公室或网络教育高校出具的相关证明。
- (8) 对于少数民族骨干计划考生, 须查验是否具有《报考 2019 年少数民族高层次骨干人才计划硕士研究生考生登记表》。没有《登记表》的, 不能列入少数民族骨干计划。
- (9) 对于“退役大学生士兵”专项计划考生, 须查验是否具有《入伍批准书》和《退出现役证》。

2. 在资格审查时, 考生须携带上述材料原件和复印件各一套, 复印件由院系留存。经审查, 若发现考生不符合报考条件, 立即取消其复试资格。

五、复试的内容、时间地点及方式

复试包括专业的综合知识与技能、外语水平及思想政治品德等内容, 同时重视本科阶段学习情况和科研能力。复试内容包括上机考试和面试。每位考生的面试复试时间一般不少于 20 分钟, 其中外语口试 5 分钟, 面试阶段全程录音。同等学力考生需加试两门科目:《数据结构》和《计算机系统基础》, 闭卷考试; 加试成绩不合格者, 不予录取。

机考的考试时间为 150 分钟, 闭卷考试, 不能携带任何形式的资料。考试内容为程序设计、数据结构和算法, 考试环境: Java, C, C++。

复试时间:3 月 21 日和 22 日, 地点: 张衡路 825 号复旦大学张江校区, 具体安排在研究生报考服务系统中通知。

六、复试评分标准

考生的总成绩包括初试成绩和复试成绩, 初试成绩和复试成绩各占 50%。

复试成绩中, 上机考试成绩占 40%, 计算机专业知识面试成绩占总成绩的 50%, 英语面试成绩占总成绩的 10%。复试成绩不及格者, 不予录取。

七、分组面试

本次复试, 学术学位单独一组复试, 专业学位分组依据研究生初试成绩采用蛇形分组模式, 分 2 组复试。

本次学术学位研究生招生名额除原来计划名额外, 另外增加了与复旦大学人类表型组研究院联合培养的名额; 专业学位招生名额中包括普通专业学位和与爱尔兰都柏林大学联合培养专业学位两类。复试前考生在学院规定的时间内根据上述情况填写志愿。

八、考生咨询方式和监督与申诉渠道

考生咨询方式: 软件学院研究生教务办公室, 联系电话: 021-51355555-11, 邮箱: zhangzhiyuan@fudan.edu.cn。

监督与申诉渠道: 学院党委办公室, 联系电话: 021-51355555-28, 邮箱: cs_dangban@fudan.edu.cn。

本细则自公布之日起实施, 由软件学院负责解释。

复旦大学软件学院
2019 年 3 月 12 日

2019 年复旦大学软件学院软件工程（非全日制）专业学位硕士研究生招生复试实施细则

根据复旦大学制定的《复旦大学 2019 年硕士研究生招生复试录取办法》，软件学院特制定《复旦大学 2019 年软件学院软件工程（非全日制）专业学位硕士研究生招生复试实施细则》。

（一）复试名单的确定

由学院研究生招生工作领导小组依据学校公布的考生参加复试的基本要求、专业招生计划和考生的初试成绩择优拟定，初试成绩达标的考生全部进入复试。复试考生名单在我院网站上公布。

（二）招生领导小组和复试小组的构成

学院高度重视复试工作，加强领导，精心组织。由学院院长、分管研究生工作的教学院长、纪检委员等组成学院招生领导小组。选派责任心强、教学经验丰富、学术及外语水平较高的老师组成复试小组，复试小组成员一般不少于 5 人。复试小组成员会认真贯彻“公平公正，客观准确”原则，严格把好复试的质量关，并严格执行回避制度。

复试小组负责确定复试的内容，负责复试命题和评定复试成绩，对考生写出评语并提出是否录取的意见。

（三）招生监察

学院党委选派合适人员承担招生监察工作，安排纪检人员或指定专人列席面试，对复试工作实施全程监督和重点时段监督。

（四）复试资格审查

所有考生的报考资格将在复试时进行审查，届时请带好以下材料

的原件和复印件,复印件需交院系存留,经审查,若发现考生不符合报考条件,立即取消其复试资格,初试成绩无效:

(1) 考生有效身份证件原件。

(2) 应届生的学生证原件。

(3) 往届生的学历证书原件和教育部学历证书电子注册备案表(或学历认证报告),学位证书原件(报考对学位有要求专业的考生)。

(4) 持在境外获得学历/学位的考生,须提交教育部留学服务中心出具的认证报告(境外学校在读本科生,不能以应届生身份报考)。

(5) 对于同等学力考生,需查验授课学校教务部门提供的8门相关专业本科课程考试成绩单原件。

(6) 对于在读研究生,需查验培养单位出具的同意报考证明;拟录取前考生还需于提供注销原学籍证明。

(7) 对于在2019年9月1日前可取得国家承认本科毕业证书的自学考试和网络教育本科生,需查验颁发毕业证书的省级高等教育自学考试办公室或网络教育高校出具的相关证明。

(8) 对于少数民族骨干计划考生,需查验是否具有《报考2019年少数民族高层次骨干人才计划硕士研究生考生登记表》。没有《登记表》的,不能列入少数民族骨干计划。

(9) 对于“退役大学生士兵”专项计划考生,需查验是否具有《入伍批准书》和《退出现役证》。没有这两项材料的考生,不能列入退役大学生士兵计划。

(五) 复试的内容、时间及地点及方式

复试采用面试的形式，主要考察学生对计算机综合知识的掌握和理解能力，应用专业知识分析和解决实际问题的能力等方面。每位考生的复试时间一般不少于 20 分钟，其中英语口语试约 5 分钟，进行全程录音。复试全程考生不得使用手机等通讯设备。复试内容具体包括如下方面：

(1) 专业知识与技能复试：

对参与复试考生，着重考核其应用专业知识分析和解决实际问题的能力，其专业基础是否扎实，以及了解其工作中的成果等。专业知识与技能复试采取口试方式，按百分制评分。

(2) 英语口语复试：

英语采用口试方式测试口语，按百分制评分。

(3) 同等学力考生加试：

同等学力考生加 2 门本科专业主干课作为复试科目，考试科目为《数据结构》和《计算机系统基础》，采用闭卷笔试方式，分别按百分制评分。同等学力考生加试科目有一门科目笔试成绩未达 60 分的，不予录取。加试时间地点另行通知。

(4) 复试时间以及地点如下：

2019 年 3 月 24 日（周天）上午 8:30-11:30，下午 13:00-17:00。地点为：复旦大学邯郸校区（邯郸路 220 号）光华楼西辅楼三楼。

(5) 复试评分标准：

考生的总成绩包括初试成绩和复试成绩，初试成绩和复试成绩各占 50%。

复试成绩中, 专业知识与技能复试成绩占总成绩的 80%, 英语口语复试成绩占总成绩的 20%。复试成绩不合格的, 不予录用。

(六) 调剂

软件工程(非全日制)专业学位硕士研究生不接收调剂。

(七) 录取

(1) 学院按考生的总成绩进行排名; 总成绩并列的, 按复试成绩进行排名。学院择优录取, 复试成绩合格的, 可向学校申请增加名额。确定拟录取名单后, 报学校审核, 拟录取名单以学校公示的名单为准。

(2) 顺利通过政治审查后, 方可正式录取, 寄发录取通知书。

(3) 考生因报考硕士研究生与所在单位产生的问题由考生自行处理。若因此造成考生不能复试或无法录取, 招生单位不承担责任。

(八) 考生咨询方式和监督与申诉渠道

考生咨询方式: 学院在职工程硕士综合管理办公室, 联系电话: 021-55664336-812, 邮箱: fudanmezhaosheng@fudan.edu.cn。

监督与申诉渠道: 学院党委办公室, 联系电话: 021-51355555-28, 邮箱: cs_dangban@fudan.edu.cn。

本细则自公布之日起实施, 由复旦大学软件学院负责解释。

复旦大学软件学院

2019 年 3 月 12 日

复试经验

2017【一】

初试结束以后就自己爽了一个月，一直到放假回家，然后回家又爽了一个月。。。大约2月中旬要回上海的时候开始准备的复试，对初试一直蛮有信心的，复试分为上机和面试两个环节。

上机我是用的王道算法指南和后来丁神推荐我的算法笔记复习的，九度+pat+codeup 大概一共刷了 180 道左右的题目吧。一开始是跟着王道走熟悉各种算法的实现方法，后来用算法笔记补充了一下一些特殊算法和模板题，机试前半周就没怎么做新的题目了，一直在看过去的算法和背一些模板题。推荐各位一开始的时候就用 vs2008 来写，我机试准备的时候一直用 vs2015 做的，后来到考场上的时候用 vs2008 真的不适应，一开始 a+b 都提交了好几次。

机试的题目各位也都看过了，题目不难，都是模板题，难在 case 上，这就要各位在考场上放平心态了，我就是很多简单题一直 a 不了所以心态有点爆炸，最后才做了 230 分。

英文面试准备了自我介绍，准备了很多，老师也是根据你自我介绍来问的，我自我介绍提前准备了一些之前做过的项目，所以老师也简单问了几句，我都准备过回答上来了，老师发音蛮清楚的，一般考研好好背过单词的应该都能听懂。

中文面试也是先让你自我介绍，然后根据自我介绍来讲，我自我介绍也是讲了我本科做的一个 SIFT 算法的图像识别和毕设的卷积神经网络的一些东西，感觉理论的老师除了我提前找的导师以为其他的老师都没有做过这个，所以老师聊得比较少，但是就是简单的问一些关于深度学习的算法的认识，我全程盯着章忠志大佬，生怕他问我汉诺塔问题。因为是提前联系过导师的，所以老师们还是蛮放我一马的感觉，就是老师一个嫌弃我用的深度模型太简单，我回答说学校没有 GPU 给我用，一个嫌弃我为什么不能报研，我说本科大一到大二都去搞学生会和辩论了。我面试的思路就是在自我介绍的时候把每一个老师可能问的点都想清楚怎么回答，让老师跟着你的思路走，老师没有打断你自我介绍的话那回答的很多问题都是提前准备过的，会舒服很多。总的来讲复旦老师真的面试都很和蔼的，特别是金玲飞老师简直女神风采，女神的长相给人的压力小的很多，听其他组的同学面试都会有很不错的老师，甚至听说媒体陈雁秋老师会当场给你解释原理，体验是很不错的。

各位如果未来想准备项目，做图像方向的话，可以推荐各位看一下 [tensorflow.org](https://www.tensorflow.org) 里面有介绍一个手写字符体识别的实现，现在深度学习框架很多，无脑一点用个 keras 实现一个模型难度也不大，面试的时候弄清楚原理吹一吹也不是很难。

今年复旦一扩再扩，最后学专考研一共收了 190+ 个人吧，总体来说还是蛮良心的，虽然出什么都慢吞吞的但是也是一种内心的磨砺吧！希望学弟学妹们不要紧张，喜欢上海或者喜欢复旦就考就是了！

2017【二】

3月27, 28 两天复试

27 上午是机试，我 AC 了一个（比较菜），算 case 的话 30 个我过了 25 个。

27 下午是英语口语面试，就背了自我介绍，老师人很好，和他们随使用英语聊聊天的感觉。

28 专业课面试就说了一个自己做的项目然后老师问几个问题，这个比较随机。

还有联系导师很重要哦！！

2018【一】

一定要趁早准备，感觉跨考的小伙伴如果有时间一定要做个项目出来，不然面试的时候真的只能尬聊了。今年复旦的 OJ 变了，看不到结果，甚至连自己是否提交成功都不知道。。不过还好题目不难，树图都没考。我在复试前差不多刷完了 PAT 乙级，甲级刷了一半，机试题目做的很顺利。还有，一定要写博客，记录 STL 使用方法和如何调 BUG。英文面试比较坑，让我介绍 campus，结果我一直介绍学校的历史，后来老师让我介绍机电一体化（我想说我是材料成型而非机电方向），也是吭哧了半天，也就最后问我为何考计算机回答的好一些。专业面试的时候就比较尴尬，老师问我对 CV 有什么认识，问我有没有项目，回答的都不好。反正大家一定要做个项目，千万不要给老师留下这个学生只会考试的印象。

2018【二】

在复试成绩刚出来的时候，可以联系老师，但要注意老师不会立刻就回的，但一般两天没回，那就可以继续联系其他老师了。也可以同时联系多位老师，但一定要注意，联系的老师不能是同一方向的老师，一定要注意。

成绩刚出来的时候，在各个群里会有人带节奏的，说分数线要多少多少了，今年的人数炸了之类的消息，不要理会，好好准备复试的机试和面试，复试群里会有组织排名的，那个虽说仅供参考，但是一般差率在 20+- 甚至 10+- 以内，可以作为重要的参考。

要记住，复试之中的机试和面试都重要，不要想着某个可以放一下，都要好好准备的。机试零分的，老师就算再想要你，也是要面试的时候表现极好，且是之前联系过某位老师，通过了该老师的电面之类的测试。

到复试通知下来的时候，如果此时还有之前多位老师在联系你，你要选择一位你想跟的老师，跟其他老师说明白，你想跟哪位老师，老师也已经说复试过之类就要你之类的，要礼貌委婉。机试继续准备着，面试可以是事先准备好的一些项目，也可以是对你联系老师的研究方向的一些了解，但绝不能说自己有了深刻理解之类的，要事实求是，会就会，不会就不会，坚决不能不懂装懂。

复试通知下来后，更要好好准备复试机试和面试了。

准备机试，不论是有无基础都可以在 pat 网上刷题的，无基础的可以先从乙级刷起，一定要在初试结束后就开始刷题，不然对于那些没参加过 acm，又不常编程练手的同学会很吃亏。机试题很大程度上在与熟练度的。还有一点，一定要注意，在做 pat 的时候，最好写做题笔记，因为有些题，你在做的时候可能不会做，但当你做题多了之后，再回去做之前不会的题，可能就很容易了。还有笔记是很好的复习方式。

如何准备面试，我自己也说不好，但有几个高端的项目就很好，就是涉及一些算法的项目，没有项目的话，或没时间做项目的话，那就把自己大学最擅长的课程，回顾一下，找一下最擅长的。最好学习一下自己选的导师的研究方向，一定要能在面试的时候说出来，但要注意措辞，不能说很熟悉之类的话，**要谦虚，谨言，慎行**。

现在我就说一下我的复试经历吧：

在这一届，先进行的是机试，而扩不扩招，会在机试前给大家说的，一个 ppt 投影各个方向的名额，在机试前要填写方向志愿的，一共三个，成绩不自信的同学，最好把一志愿填大家都不怎么会把该方向作为一志愿的，因为这样录取机会极大，而选择了人数填报多的方向就竞争比较大了，这要自己去斟酌。因为我之前联系了导师，一志愿就填的导师所在的方向，而二志愿和三志愿都填的人数比较多的。本届机试的提交是什么结果都不显示的，但题比较水，对于我就是要么 ak，要么 0 分了，从最后老师说我被学院录了的情况看，我是 ak 的。估计因为提交看不了结果，case 数据很弱，所以很容易就 ak 的。一些机试零分的，原本可能也是 ak，但因为马虎，编译没有通过，又看不到提交情况，就机试零分了。

机试的情况别想以某些年为例子了，因为旦旦没有套路的，老老实实刷机试题就对了。

在机试结束后，给大家一个半小时的吃饭时间，接着就到了下午的英语面试，英语面试要看人品了，而且看的是你爸妈留给你的姓氏的人品，因为有些英语面试的老师很严格，有些英语面试的老师就很和蔼。我的人品不错，面试我的时候，女老师出去打电话了，男老师人很好，一直让我自我介绍，然而，我自我介绍准备的太少了。因为我原本就想要的是那种问答形式的英语面试，所以在准备的自我介绍讲完后，老师又不问问题，我只能在那找话说。所以，大家要注意，自我介绍一定要准备的多一些。接着就是老师根据

我自我介绍里的内容提问了，因此，在自我介绍了千万别讲一些自己不懂，或是懂但用英文讲不出来的东西。碰见严格的老师，只能看你自己的知识储备和心理素质了。

在第二天进行的专业知识面试，是根据自己选的方向分组面试的，**选方向的时候一定要注意**，如果都选的人很多的方向，自己很可能排不到面试，该方向就满员不招了。

我很幸运，三个志愿方向都面完了。关于面试，还是要认真准备面试的，要在自己擅长的部分准备，没有擅长的部分，那就找选的老师的研究方向的一些学习情况，但这类，一定要注意不能把话说得很高，要谦卑，因为面试的老师都是业界大佬，一定要注意。可以卖弱，但过犹不及，我就卖弱太过，使得面试的时候表现不好，但幸好，我面试组的老师给分高一些。

在复试结束后就是等待消息的时间了，旦旦的通知会下的比较晚，要有耐心，可以通过学习，水群，玩游戏转移一下注意力，不要被一些群里带了节奏，填了调剂，说不定，你被复旦录了但因为填了调剂，就gg了。

对了，还有之前说的，可以联系多位老师，但联系的老师不要联系同一个方向的老师，因为老师们会开会决定一些学生的去留，而联系了同方向的多位老师，老师们都想帮帮学生，然后就比较尴尬了（别问我怎么知道的~。~）。

一般在一个星期里，被一志愿要的学生都能知道自己是否被录，不知道的一种是老师也不确定是否去留，要开会的讨论才能决定的，另一种就是被一志愿刷了，等学院通知就好了，旦旦刷人比例还好，大都会有学上的，就是方向要到拟录取邮件发过来才知道的。

2018【三】

复旦大学工研院计算机复试（流程+机考题目）

复试有机考和面试两部分。

上午机考：

考试时间两个小时，C/C++编译环境有 VS2007、VS2010、codeblock，Java 环境也有，编译器没注意，欢迎补充，也可参看计算机学院的学长介绍，机考在张江校区计算机楼进行。

OJ 系统也和计算机学院是一样的（HUSTOJ），上传之后不显示分数，循环处理多个用例。一共三道题，每道题包括一个大测试集和一个小测试集，所以实际上需要提交 6 道题。**注意提交的时候会选择 C/C++/Java。**

机考题目回顾：

第一题，输入两个集合，分别求其交集和并集中元素的个数，每个集合中可能存在相同的元素，而最终的交集和并集中应该不存在。

例如：

输入：

```
4 5
3 4 7 3
4 6 3 2 6
```

输出：

```
2 5
```

第二题，输入一个数 n ，输出前 n 个数的约数的和。（印象中有 1s 的时间限制，大数据集可能超时，比如 100000000）。

例如：

输入：

```
7
```

输出：

```
41
```

第三题，求线段交点，输入两组线段端点（整型），求其交点，不相交和无穷交点输出一句话就行，输出交点带小数的。

例如：

输入：

```
0 0 5 5
0 2 2 0
```

输出：

```
1 1
```

下午面试：

按名单顺序，先进行专业课面试，再到另一个房间进行英语面试，面试主要依据你的个人简历展开

