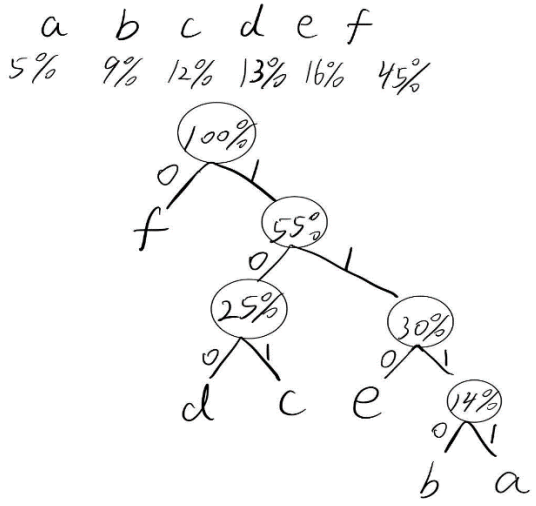
CSP-J 考点汇总

1.逻辑计算：&&和 ||或 ！非
2.位运算

2. 位运算概览

符号	描述	运算规则
&	与	两个位都为1时，结果才为1
|	或	两个位都为0时，结果才为0
^	异或	两个位相同为0，相异为1
~	取反	0变1，1变0
<<	左移	各二进位全部左移若干位，高位丢弃，低位补0
>>	右移	各二进位全部右移若干位，高位补0或符号位补齐

3.进制转换
N 进制转十进制—>按位乘法
十进制转 N 进制—>短除法，从下往上取数
N 进制含小数转十进制—>整数转换+按小数位乘法
十进制含小数转 N 进制—>整数转换+小数部分竖式乘法按正常顺序取进位
4.编码:
ASCII 码：略
哈夫曼编码：（贪心思想）
假设有一组字符 {a,b,c,d,e,f}，对应的频率分别
为 5%,9%,12%,13%,16%,45%5%,9%,12%,13%,16%,45%。请问以下哪个选项是字符 abcdef 分别对应的一组哈夫曼编码?
答案：1111,1110,101,100,110,0
解析过程：根据频率构造哈夫曼树（sort）



按频率总和 100%，分配左右子树，以 a，e,d 为例，左边写 0，右边写 1，则 a 就是 1111,e 是 110,d 是 100

5.排列组合:

加法原理：做一件事有几种方案

乘法原理：一件事有几个步骤

$$A_n^m = \frac{n!}{(n-m)!} \qquad C_n^m = \frac{n!}{m!\,(n-m)!}$$

（2）解题技巧

技巧 1：相邻问题——整体捆绑法 。

例：7 名学生站成一排，甲、乙必须站在一起，有多少不同排法?

解：两个元素排在一起的问题可用“捆绑”法解决，先将甲乙二人看作一个元素与其他五人进行排列，并考虑甲乙二人的顺序，所以共有 A(6/6) * A(2/2) =1440 种。

例：5 个男生 3 个女生排成一排，3 个女生要排在一起，有多少种不同的排法?

解：因为女生要排在一起，所以可以将 3 个女生看成是一个人，与 5 个男生作全排列，因此有 A(6/6) 种排法，其中女生内部也有 A(3/3) 种排法;

根据乘法原理，共有 A(3/3) * A(6/6) = 4320 种不同的排法。

技巧 2：不相邻问题——选空插入法

例：7 名学生站成一排，甲乙互不相邻，有多少不同排法?

解：甲、乙二人不相邻的排法一般应用“插空”法，所以甲、乙二人不相邻的排法总数计算应为： A(5/5)*A(2/6) =3600 种。

例：学校组织老师学生一起看电影，同一排电影票 12 张，8 个学生，4 个老师，要求老师在学生中间，且老师互不相邻，共有多少种不同的坐法？（写出计算公式，不需要计算）

解：先排学生共有 A(8/8) 种排法，然后把老师插入学生之间的空档，共有 7 个空档可插，选其中的 4 个空档，共有 A(4/7) 种选法。根据乘法原理，共有的不同坐法为 A(8/8) * A(4/7) = 33868800 。

技巧 3：复杂问题——总体排除法或排异法

例：正六边形的中心和顶点共7个点，以其中3个点为顶点的三角形共有_____个。

解：从 7 个点中取 3 个点的取法有 C(3/7) 种，但其中正六边形的对角线所含的中心和顶点三点共线不能组成三角形，有 3 条，所以满足条件的三角形共有 C(3/7)-3=32 个。

例：从 43 人中任抽 5 人，正、副班长、团支部书记至少有一人在内的抽法有多少种?

解：43 人中任抽 5 人的方法 C(5/43) 种，正副班长，团支部书记都不在内的抽法有 C(5/40)种，所以正副班长，团支部书记至少有 1 人在内的抽法有 C(5/43)-C(5/40) 种。

技巧 4：特殊元素——优先考虑法

比特顽童ACM教练团 内部参考资料请勿外传！

例：乒乓球队的 10 名队员中有 3 名主力队员，派 5 名队员参加比赛，3 名主力队员要安排在第一、三、五位置，其余 7 名队员选 2 名安排在第二、四位置，那么不同的出场安排共有_____种。

解：由于第一、三、五位置特殊，只能安排主力队员，有 A3/3 种排法，而其余 7 名队员选出 2 名安排在第二、四位置，有 A2/7 种排法，所以不同的出场安排共有 A3/3*A2/7=252 种。

技巧 5：多元问题——分类讨论法

例：某班新年联欢会原定的 5 个节目已排成节目单，开演前又增加了两个新节目；如果将这两个节目插入原节目单中，那么不同插法的种数为_____。

解：增加的两个新节目，可分为相邻与不相邻两种情况

(1) 不相邻：共有 A2/6 种；

(2) 相邻：共有 A2/2*A1/6 种。故不同插法的种数为： 30+12=42 ；

该题的另外一个解法为：

(1) 7 个节目的总排法为 A(7/7) ；

(2) 但其中 5 个节目是预先排好的，排法为 A(5/5) ；

(3) 那么剩余的 2 个节目的排法就要去掉这个重复的 5 个节目的排法，结果为 A(7/7) / A(5/5) = 7 * 6 = 42 ；

技巧 6：混合问题——先选后排法

例：从黄瓜、白菜、油菜、扁豆 4 种蔬菜品种中选出3 种，分别种在不同土质的三块土地上，其中黄瓜必须种植，不同的种植方法共有_____种。

解：先选后排，分步实施。由题意，不同的选法有 C(2/3) 种，不同的排法有 A(3/3) 故不同的种植方法共有 C(2/3)*A(3/3)=18 。

技巧 7：相同元素分配——档板分隔法

例：把 10 本相同的书发给编号为 1，2，3 的三个学生阅览室，每个阅览室分得的书的本数不小于其编号数，试求不同分法的种数有_____种。

解：先分别给 1、2、3 号阅览室分配 0、1、2 本书，然后把剩下的 7 本书分配给这 3 个阅览室，要求每个阅览室至少再分到 1 本。那么在 7 本书中有 6 个空，任意选取 2 个空，就可以把这 7 本书分成左、中、右 3 个部分，分别分给 3 个图书室，所以不同的分法共有 C2/6 = 15 。

技巧 8：转化法

例： 高二年级 8 个班，组织一个 12 个人的年级学生分会，每班要求至少 1 人，名额分配方案有_____种？

转化法：对于某些较复杂的或较抽象的排列组合问题，可以利用转化思想，将其化归为简单的、具体的问题来求解。

解：此题若直接去考虑的话，就会比较复杂。但如果我们将其转化为等价的其他问题，就会显得比较清楚，方法简单，结果容易理解。此题可以转化为：插板问题，结果为 C7/11=330 。

6.计算日期算法

1. 今天是星期日，那么 100 天以后是星期几？

解析：

本题是周期性问题求解，每周有 7 天；

那么 100 天除去完整的 7 天周期以外，剩余填数 = 100 % 7 = 2（14 个完整周期剩余 2 天）；

因此 100 天以后是星期几，相当于 2 天之后是星期几，答案是星期二。

2、今天是星期日，那么 10^{100} 天以后是星期几？

解析：

我们并不急于求出 10^{100}，而是像 1, 10, 100, 100, 10000…这样，依次增加 0 的个数， 观察其规律。

0 的个数			
0	1 天以后的星期数	1÷7=0 余 1	-> 一
1	10 天以后的星期数	10÷7=1 余 3	-> 三
2	100 天以后的星期数	100÷7=14 余 2	-> 二
3	1000 天以后的星期数	1000÷7=142 余 6	-> 六
4	10000 天以后的星期数	10000÷7=1428 余 4	-> 四
5	100000 天以后的星期数	100000÷7=14285 余 5	-> 五
6	1000000 天以后的星期数	1000000÷7=142857 余 1	-> 一
7	10000000 天以后的星期数	10000000÷7=1428571 余 3	-> 三
8	100000000 天以后的星期数	100000000÷7=14285714 余 2	-> 二
9	1000000000 天以后的星期数	1000000000÷7=142857142 余 6	-> 六
10	10000000000 天以后的星期数	10000000000÷7=1428571428 余 4	-> 四
11	100000000000 天以后的星期数	100000000000÷7=14285714285 余 5	-> 五
12	1000000000000 天以后的星期数	1000000000000÷7=142857142857 余 1	->一

因此 10^{100} 以后的星期数，可以将天数中的 0 的个数（10^{100} 有 100 个 0）除以 6，通过所得余数来判断。

100 ÷ 6 = 16 余 4

因此，10^{100} 天以后是星期四。

3. 2017 年 10 月 1 日是星期日，1999 年 10 月 1 日是（）。

A. 星期三　　B. 星期日　　C. 星期五　　D. 星期二

解析：

闰年：一年 366 天，非闰年：一年 365 天。

判断标准：四年一闰，百年不闰，四百年再闰（n%4==0&&n%100!=0 ||n % 400 == 0）

意思是：不是整百的年份只要被 4 整除的就是闰年，整百的年份必须得被 400 整除才是闰年

1999 年~2016 年中的闰年有：2000、2004、2008、2012、2016，5 个闰年。

也就是说 1999 年 10 月 1 日~2017 年 10 月 1 日总天数 = 18 年，总天数 = 18 * 365 + 5 = 6575 天。6575/7 结果为 939 周余 2 天。

说明：1999 年 10 月 1 日为周五。

7.前缀、中缀、后缀转换/运算

6.4 前缀、中缀、后缀表达式

前缀表达式、中缀表达式、后缀表达式都是四则运算的表达方式，用以四则运算表达式求值，即数学表达式的求值。**中缀表达式**就是常见的运算表达式，如(3+4)×5-6。

前缀表达式又称波兰式，前缀表达式的运算符位于操作数之前，比如：- × + 3 4 5 6。

前缀表达式和后缀表达式都符合计算机的逻辑，方便计算机计算数据。

1、前缀表达式

A、前缀表达式求值

从右至左扫描表达式，遇到数字时，将数字压入堆栈，遇到运算符时，弹出栈顶的两个数，用运算符对它们做相应的计算（栈顶元素 top 和 次顶元素），并将结果入栈；重复上述过程直到表达式最左端，最后运算得出的值即为表达式的结果。

例如前缀表达式：- × + 3 4 5 6 求值过程如下：

（1）**从右至左扫描**，将 6、5、4、3 压入堆栈

（2）遇到+运算符，因此弹出 3 和 4（3 为栈顶元素，4 为次顶元素，注意与后缀表达式做比较），计算出 3+4 的值，得 7，再将 7 入栈

（3）接下来是×运算符，因此弹出 7 和 5，计算出 7×5=35，将 35 入栈

（4）最后是-运算符，计算出 35-6 的值，即 29，由此得出最终结果

【NOIP2010 提高组】前缀表达式“+3*2+5 12”的值是（　　）

A. 23　　B. 25　　C. 37　　D. 65

答案：C

B、将中缀表达式转换为前缀表达式

转换过程需要用到栈，具体过程为：从右向左读取表达式

1）如果遇到操作数，我们就直接将其输出。

2）如果遇到操作符，则我们将其放入到栈中，遇到右括号时我们也将其放入栈中。

3）如果遇到一个左括号，则将栈元素弹出，将弹出的操作符输出直到遇到右括号为止。注意，右括号只弹出并不输出。

4）如果遇到任何其他的操作符，如（“+”，“*”，“（”）等，从栈中弹出元素直到遇到发现**相同或更低**优先级的元素(或者栈为空)为止。弹出完这些元素后，才将遇到的操作符压入到栈中。有一点需要注意，只有在遇到”(”的情况下我们才弹出”)”，其他情况我们都不会弹出”）”。

5）如果我们读到了输入的末尾，则将栈中所有元素依次弹出。

6）最后，将弹出的结果，再次逆序，得到最终的结果。

例如：1+((2+3)×4)-5 转换为前缀表达式，结果是：　- + 1 × + 2 3 4 5

1+2*3+(4*5+6)*7 转换为前缀表达式，结果是：＋＋１＊２３＊＋＊４５６７

2、后缀表达式

后缀表达式又称逆波兰表达式，与前缀表达式相似，只是运算符位于操作数之后；比如：3 4 + 5 × 6 -

A、后缀表达式计算机求值

与前缀表达式类似，只是顺序是从左至右：

从左至右扫描表达式，遇到数字时，将数字压入堆栈，遇到运算符时，弹出栈顶的两个数，用运算符对它们做相应的计算（次顶元素 和 top 栈顶元素），并将结果入栈；重复上述过程直到表达式最右端，最后运算得出的值即为表达式的结果

例如后缀表达式“3 4 + 5 × 6 -”。

从左至右扫描，将 3 和4 压入堆栈；

遇到+运算符，因此弹出 4 和 3（4 为栈顶元素，3 为次顶元素，注意与前缀表达式做比较），计算出 3+4 的值，得 7，再将 7 入栈；

将 5 入栈；

接下来是×运算符，因此弹出 5 和 7，计算出 7×5=35，将 35 入栈；

将 6 入栈；

最后是-运算符，计算出 35-6 的值，即 29，由此得出最终结果。

B、将中缀表达式转换为后缀表达式

转换过程需要用到栈，具体过程如下，从左向右扫描表达式：

1）如果遇到操作数，我们就直接将其输出。

2）如果遇到操作符，则我们将其放入到栈中，遇到左括号时我们也将其放入栈中。

3）如果遇到一个右括号，则将栈元素弹出，将弹出的操作符输出直到遇到左括号为止。注意，左括号只弹出并不输出。

4）如果遇到任何其他的操作符，如（“+”，“*”，“（”）等，从栈中弹出元素直到遇到发现更低优先级的元素(或者栈为空)为止。弹出完这些元素后，才将遇到的操作符压入到栈中。有一点需要注意，只有在遇到”）“的情况下我们才弹出”（“，其他情况我们都不会弹出”（“。

5）如果我们读到了输入的末尾，则将栈中所有元素依次弹出。

例如，中缀表达式 1+((2+3)×4)-5 转换为后缀表达式的结果为：123+4×+5-

中缀表达式 a+b*c+(d*e+f)*g，其转换成后缀表达式则为 abc*+de*f+g*+

8.排序算法：

排序方法	时间复杂度（平均）	时间复杂度（最坏）	时间复杂度（最好）	空间复杂度	稳定性
插入排序	$O(n^2)$	$O(n^2)$	$O(n)$	$O(1)$	稳定
希尔排序	$O(n^{1.3})$	$O(n^2)$	$O(n)$	$O(1)$	不稳定
选择排序	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(1)$	不稳定
堆排序	$O(nlog_2n)$	$O(nlog_2n)$	$O(nlog_2n)$	$O(1)$	不稳定
冒泡排序	$O(n^2)$	$O(n^2)$	$O(n)$	$O(1)$	稳定
快速排序	$O(nlog_2n)$	$O(n^2)$	$O(nlog_2n)$	$O(nlog_2n)$	不稳定
归并排序	$O(nlog_2n)$	$O(nlog_2n)$	$O(nlog_2n)$	$O(n)$	稳定
计数排序	$O(n+k)$	$O(n+k)$	$O(n+k)$	$O(n+k)$	稳定
桶排序	$O(n+k)$	$O(n^2)$	$O(n)$	$O(n+k)$	稳定
基数排序	$O(n*k)$	$O(n*k)$	$O(n*k)$	$O(n+k)$	稳定

1. **冒泡排序（Bubble Sort）**
   - **原理**：
     - 比较相邻的元素。如果第一个比第二个大（升序），就交换它们两个；
     - 对每一对相邻元素作同样的工作，从开始第一对到结尾的最后一对，这样在最后的元素应该会是最大的数；
     - 针对所有的元素重复以上的步骤，除了最后已经选出的最大元素；
     - 持续每次对越来越少的元素重复上面的步骤，直到没有任何一对数字需要比较。
   - **示例**：
     - 对数组[5, 4, 3, 2, 1]进行冒泡排序。
     - 第一轮比较：比较 5 和 4，交换得到[4, 5, 3, 2, 1]；比较 5 和 3，交换得到[4, 3, 5, 2, 1]；以此类推，第一轮结束后数组变为[4, 3, 2, 1, 5]。
     - 第二轮比较：对前 4 个元素[4, 3, 2, 1]进行类似操作，结束后数组变为[3, 2, 1, 4, 5]。
     - 经过多轮比较，最终数组变为[1, 2, 3, 4, 5]。

2. **插入排序（Insertion Sort）**
   - **原理**：
     - 将待排序的元素插入到已经有序的部分合适位置。
     - 从第二个元素开始，将其与前面已排序的元素依次比较，找到合适的位置插入，使得插入后仍然有序。
   - **示例**：
     - 对数组[5, 4, 3, 2, 1]进行插入排序。
     - 从第二个元素 4 开始，4 与 5 比较，4 小于 5，将 4 插入到 5 前面，得到[4, 5, 3, 2, 1]。
     - 接着处理元素 3，3 依次与 5、4 比较，将 3 插入到合适位置，得到[3, 4, 5, 2, 1]。
     - 以此类推，最终数组变为[1, 2, 3, 4, 5]。

3. **选择排序（Selection Sort）**
   - **原理**：
     - 首先在未排序序列中找到最小（大）元素，存放到排序序列的起始位置；
     - 然后，再从剩余未排序元素中继续寻找最小（大）元素，然后放到已排序序列的末尾；
     - 重复第二步，直到所有元素均排序完毕。
   - **示例**：
     - 对数组[5, 4, 3, 2, 1]进行选择排序。
     - 第一轮找到最小元素 1，与第一个元素 5 交换，得到[1, 4, 3, 2, 5]。
     - 第二轮在剩余元素[4, 3, 2, 5]中找到最小元素 2，与第二个元素 4 交换，得到[1, 2, 3, 4, 5]。
     - 继续这个过程直到数组有序。

4. **快速排序（Quick Sort）**
   - **原理**：
     - 选择一个基准元素，通常选择数组的第一个元素或者最后一个元素；
     - 通过一趟排序将待排序列分割成两部分，其中一部分的所有元素都比基准元素小，另一部分的所有元素都比基准元素大；

- 然后分别对这两部分递归地进行快速排序。

- **示例**：
  - 对数组[5, 4, 3, 2, 1]进行快速排序，选择 5 作为基准元素。
  - 经过一趟排序后，数组变为[1, 4, 3, 2, 5]，其中左边部分[1]都比 5 小，右边部分[4, 3, 2]都比 5 大；
  - 接着对[1]和[4, 3, 2]分别进行快速排序，直到整个数组有序。

5. **归并排序（Merge Sort）**
   - **原理**：
     - 把长度为 n 的输入序列分成两个长度为 n/2 的子序列；
     - 对这两个子序列分别采用归并排序；
     - 将两个排序好的子序列合并成一个最终的排序序列。
   - **示例**：
     - 对数组[5, 4, 3, 2, 1]进行归并排序。
     - 首先分成[5, 4]和[3, 2, 1]；
     - 对[5, 4]再分[5]和[4]，排序后合并为[4, 5]；
     - 对[3, 2, 1]再分[3]、[2, 1]，[2, 1]再分[2]和[1]，排序后合并为[1, 2]，再与[3]合并为[1, 2, 3]；
     - 最后将[4, 5]和[1, 2, 3]合并为[1, 2, 3, 4, 5]。

6. **希尔排序（Shell Sort）**
   - **原理**：
     - 先将整个待排序的记录序列分割成为若干子序列分别进行直接插入排序；
     - 选择一个增量序列，通常是递减的，如希尔增量序列（n/2, n/4, n/8,...）；
     - 按照增量序列依次对各子序列进行排序，当增量减至 1 时，整个序列恰被分成一个子序列，排序完成。
   - **示例**：
     - 对数组[5, 4, 3, 2, 1]进行希尔排序，设增量序列为[3, 1]。
     - 当增量为 3 时，分成子序列[5, 2]、[4, 1]、[3]，分别进行插入排序后得到[2, 1, 3, 5, 4]；
     - 当增量为 1 时，对整个数组进行插入排序得到[1, 2, 3, 4, 5]。

7. **基数排序（桶排序）**
   - **原理**：
     - 将整数按位数切割成不同的数字，然后按每个位数分别比较。
     - 从最低有效位开始，依次进行排序，直到最高有效位。
   - **示例**：
     - 对数组[123, 456, 789, 132, 465]进行基数排序。
     - 先按个位数字进行排序，得到[132, 123, 465, 456, 789]；
     - 再按十位数字排序，得到[123, 132, 456, 465, 789]；
     - 最后按百位数字排序，数组保持不变。

只有基数（桶排序）排序不需要进行关键字比较，其余排序算法皆需要进行。

9.数据结构
栈：先进后出
队列：先进先出
结构体：struct
链表：
非连续性存储的线性结构，通过指针链接元素次序。
Int *p=&x;定义指针 p 指向 x 的地址
链表中各个元素在内存中位置零散的，连续或不连续均可以，无需事先估计存储空间大小，插入、删除操作无需移动元素，但不能任意访问元素，找元素必须从第一个元素开始。
链表中不能出现一个变量已经被改掉了，然后还用这个改过的变量 这种情况。

二叉树：
有唯一的根节点，有叶子节点和结点，根节点的层数、深度、高度为 1。
二叉树度<=2
完全二叉树（必须有一个结点下方没有右儿子）/满二叉树/一般二叉树

二叉树第 i 层上最多有 2^{i-1} 个结点，深度（高度/层数）为 k 的二叉树最多有 2^k-1 个结点

扩展：深度为 k 的三叉树最多有$(3^k-1)/2$ 个结点

n0：0 度点数，n1：1 度点数，n2：2 度点数，n：该二叉树总点数
则 n0=n2+1，n=n0+n1+n2
完全二叉树中 n 个结点，那么它有 n-n/2 个叶子结点（这里 n/2 默认向下取整）。
二叉树的遍历：
前序：根-左-右 中序：左-根-右 后序：左-右-根

图：
连通图：能沿着路径抵达任何点，针对无向图。
强连通图：能演路径抵达任何点，针对有向图，注意有方向。
回路：起点和终点相连，形成类似环形的图。
完全图：任意两点间都有且仅有一条线段相连。
有向图：有方向的图
无向图：没有方向的图
公式：
有向完全图 n 个顶点，n(n-1)条边
无向完全图 n 个顶点，n(n-1)/2 条边
n 个顶点的无向连通图至少有 n-1 条边
n 个顶点的连通图，其最小生成树为 n-1 条边。
n 个点的图，其对应的树至少有 n-1 条边
n 个顶点的无向图至少应该有 n-1 条边才能保证是一个连通图
图中每个点的度等于该店入度和出度之和。
图中所有点的入度之和等于所有点的出度之和
n 个顶点的有向图，若是强连通图，则图中最少有 n 条边。

10.零散知识
如遇字符串求组合方式问题，应考虑加上空串的可能。

数据存储基本单位：Byte 字节
最小存储单位：bit 比特
1byte=8bit，1kb=1024byte，1mb=1024kb，1gb=1024mb，1tb=1024gb，1pb=1024tb
1 个 32 位 int 变量占 4 字节（byte）。

图像文件存储大小=水平像素 x 垂直像素 x 位数 / 8 最终结果单位：byte
一个汉字通常容量 2byte（16bit），16x16 的点阵汉字字模存储容量：
16x16 / 8 = 32 Byte

面向对象的编程语言：c++ 、python、java 等，汇编语言：C 等
Bool 类型计算中 尖向下符号：|| or 或 尖向上符号：&& and 和
用高级程序设计语言编写的程序必须通过编译或解释的方式翻译后才能被执行
在程序设计语言中，子程序调用语句中实际参数必须与子程序说明语句的形式参数的个数、顺序、类型保持一致
空间复杂度：执行算法所需要的内存空间

与电子邮件有关的协议：
POP3 邮局（互联网常用的 E-mail 协议）
SMTP：邮箱
IMAP：互联网

Linux 系统中可执行文件后缀“Dll”，注意不是“dll”!!!
通信软件：QQ、微信、MSN
操作系统：Solaris、Linux、Windows Vista。
Sybase 是一种数据库

IP 地址范围，任何 ip 地址上都不能出现大于 255 的数字
A 类地址:范围：0.0.0.0 - 127.255.255.255
B 类地址:范围：128.0.0.0 - 191.255.255.255
C 类地址:范围：192.0.0.0 - 223.255.255.255
D 类地址:范围：224.0.0.0 - 239.255.255.255。
E 类地址:范围：240.0.0.0 - 255.255.255.255 用途：保留用于实验和研究等特殊目的。

floor（）向下取整 ceil（）向上取整 round（）四舍五入

二分查找算法（折半查找算法）在 n 个数中查找某个元素，则最多需要查找多少次?

$2^k=n$，需要查找 k 次（k 向上取整）。

2^n 种颜色用二进制编码表示（图像），则至少有 n 位

LAN 局域网，WAN 广域网，MAN 城域网.
基于比较的排序时间复杂度下限为 O(nlogn)

一个完整的计算机系统包含：硬件系统和软件系统
矢量图：由直线和曲线构成的图像，不会失真，基于数学方程的几何图元来表示图像，但色彩不够丰富。
子串一定连续，子序列连续或不连续均可以

当只有根节点的二叉树或叶子节点，只有右子树的二叉树，它的前序遍历与中序遍历相同。当只有根节点的二叉树或叶子结点，只有左子树的二叉树，它的中序遍历和后序遍历相同。
图的深度优先遍历算法常用栈作为数据结构，栈的访问原则为后进先出（先进后出），队列常被应用于广度优先搜索算法，栈与队列存在本质不同，两个栈可以实现队列的效果，但一个不可以。