算法设计与分析练习

一、选择

1、在用计算机求解问题的过程中，最核心的问题是_______。

A、硬件配置　　B、算法的设计与实现
C、软件升级　　D、文档编制

2、下面说法中，正确的是_______。

A、算法 + 数据结构 = 程序　　B、算法就是程序
C、数据结构就是程序　　D、算法包括数据结构

3、以下工具中，_______不能用于算法的表示。

A、流程图　B、伪码　C、位示图　D、状态图

4、算法的基本控制结构不包括_______。

A、顺序结构　B、选择结构　C、循环结构　D、跳转结构

5、算法的二要素指的是_______。

A、操作与控制结构　　B、逻辑结构与存储结构
C、设计语言与开发环境　　D、数值运算与非数值运算

6、对一个算法的评价，通常不包括_______。

A、有穷性　B、正确性　C、先进性　D、可读性

7、用数量级形式表示的算法执行时间称为算法的________复杂度。

A、时间　B、空间　C、处理器　D、通信

8、渐近算法分析是指________。

A、算法在最佳情况、最差情况和平均情况下的代价
B、当规模逐步增大时，对算法资源开销“增长率”上的简化分析
C、数据结构所占用的空间
D、在最小输入规模下算法的资源代价

9、对一个算法的评价，通常不包括_______。

A、健壮性　B、并发性　C、正确性　D、运算复杂度

10、一个算法应该包含的性质，除了________。

A、二义性　B、有限性　C、正确性　D、可终止性

11、当上下限表达式相等时，使用下列的________来描述算法代价。

A、大 O 表示法　　B、大 Ω 表示法
C、Θ 表示法　　D、小 o 表示法

12、衡量一个算法好坏的标准是________。

A、运行速度快　　B、占用空间少
C、时间复杂度低　　D、代码短

13、以下程序段的时间复杂度为_______。

```
x = y = 1;
while (x <= n && y <= n)
{
    x = x + 1;
    y = y + x;
}
```

A、$O(\log_2 n)$　B、$O(\sqrt{n})$　C、$O(n)$　D、$O(n^2)$

14、以下程序段的时间复杂度为_______。

```
m = 0;
for (i = 1; i <= n; ++ i)
    for (j = 2 * i; j <= n; ++ j)
        m = m + 1;
```

A、$O(1)$　B、$O(\log_2 n)$　C、$O(n)$　D、$O(n^2)$

15、下面的算法段针对不同的自然数 n 作不同的处理，其中函数 $odd(n)$ 当 n 是奇数时返回 true，否则返回 false，

```
while (n > 1)
{
    if (odd(n))
        n = 3 * n + 1;
    else
        n = n / 2;
}
```

该算法所需计算时间的下界是________。

A、$\Omega(2^n)$　B、$\Omega(n\log n)$　C、$\Omega(n!)$　D、$\Omega(\log n)$

16、下述算法的时间复杂度为________。

```
void hanoi(int n, int a, int b, int c)
{
    if (n > 0)
    {
        hanoi(n – 1, a, c, b);
        move(a, c);
        hanoi(n – 1, b, a, c);
    }
}
```

A、$O(2^n)$　B、$O(n\log n)$　C、$\Theta(n!)$　D、$\Theta(n^n)$

17、通常使用算法的最坏运行时间来估计算法时间复杂度，下面的陈述中，________不是这样做的原因。

A、在实际问题中，算法的运行时间常常达到这个上界
B、平均运行时间难以计算
C、假设每一个输入具有相同的概率有时没有意义
D、平均运行时间与最坏运行时间有相同的数量级

18、如果用 $T(n)$ 表示当输入规模为 n 时算法的语句频度，以下算法中效率最优的是________。

A、$T(n) = T(n – 1) + 1, T(1) = 1$　　B、$T(n) = 2n^2$
C、$T(n) = T(n / 2) + 1, T(1) = 1$　　D、$T(n) = 3n\log_2 n$

19、对于下列各组 $f(n)$ 和 $g(n)$，________不满足 $f(n) = O(g(n))$。

A、$f(n) = \log n^2, g(n) = \log n + 5$　　B、$f(n) = \log n^n, g(n) = n$
C、$f(n) = n!, g(n) = n^n$　　D、$f(n) = \log n, g(n) = n$

20、$T(n)$ 表示当输入规模为 n 时的算法效率，以下算法效率最优的是________。

A、$T(n) = T(n – 1) + 1, T(1) = 1$　　B、$T(n) = 2n^2$
C、$T(n) = T(n / 2) + 1, T(1) = 1$　　D、$T(n) = 3n\log_2 n$

21、当输入规模为 n 时，算法增长率最大的是________。

A、5^n B、$20\log_2 n$ C、$2n^2$ D、$3n\log_3 n$

22、为提高算法的效率，应当优先考虑选择_______。

A、恰当的数学模型 B、高效率的编译程序

C、先进的硬件设备 D、高效率的程序设计语言

23、存储结构的选择，应当优先考虑_______。

A、算法的实现效率 B、算法设计人员的喜好

C、存储器的速度 D、算法运行所在计算机的体系结构

24、算法原地工作的含义是指_______。

A、算法的运行时间永远为常数

B、算法的运行时间相对于数据量为常数

C、算法不需要任何额外的辅助空间

D、算法需要的辅助空间相对于数据量为常数

25、穷举法的适用范围是_______。

A、一切问题 B、解的个数极多的问题

C、解的个数有限且可一一列举 D、不适合设计算法

26、通过多重循环一一列举出解决问题的所有可能解，并在逐一列举的过程中，检验每个可能的解是否是问题真正解的算法是_______，而从实际问题中归纳抽象出数学解析式，就此设计出合适的算法是_______。

A、解析法，穷举法 B、递归法，解析法

C、穷举法，解析法 D、穷举法，递归法

27、采用“顺序查找”从一个长度为 n 的随机分布数组中搜寻值为 k 的元素。以下对顺序查找分析正确的是________。

A、最佳情况、最差情况和平均情况下，顺序查找法的渐近代价都相同

B、最佳情况的渐近代价要好于最差情况和平均情况的渐近代价

C、最佳情况和平均情况的渐近代价要好于最差情况的渐近代价

D、最佳情况的渐近代价要好于平均情况的渐近代价，而平均情况的渐近代价要好于最差情况的渐近代价

28、在直角三角形中，三条边 a、b、c 的长度都为整数，且一条直角边 a 的长度已确定，斜边 c 的长度不能超过某数 i，求满足条件的所有直角三角形，采用_______方法最合理。

A、递归法 B、选择法 C、穷举法 D、解析法

29、判断某自然数 m 是否素数算法的基本思想：把 m 作为被除数，将 2 到 $m-1$ 中的所有自然数作为除数，逐一相除，如果都除不尽，m 就是素数，否则不是素数，这种判定素数的算法属于_______。

A、枚举法 B、解析法 C、递归法 D、排序法

30、将一个复杂问题的求解过程，转化为相对简单的数字计算算式的重复执行，这一类算法通称为_______。

A、回溯法 B、穷举法 C、递归法 D、迭代法

31、求代数方程根的牛顿法属于_______。

A、蛮力法 B、迭代法 C、递归法 D、回溯法

32、著名的 *Hanoi* 塔问题，是应用_______算法解决的。

A、顺序查找 B、对半查找

C、选择排序和插入排序 D、递归

33、已知分数序列 $\frac{1}{2}, \frac{2}{3}, \frac{3}{5}, \frac{5}{8}, \frac{8}{13}, \frac{13}{21}, \ldots$，如果需要设计算法求出该序列第 20 项的值，则可以使用的算法是_______。

A、贪心法 B、递推法 C、分治法 D、回溯法

34、以下关于递归与循环的陈述中，说法有错误的是_______。

A、递归算法设计中，必须要有使递归终止的条件

B、循环算法设计中，需要将重复的操作归纳为循环体

C、一般而言，等价功能的递归算法运行效率比循环算法效率高

D、借助栈，可以将递归算法变换为等价的循环算法

35、将一个正整数 n 表示成一系列正整数之和：

$n = n_1 + n_2 + \ldots + n_k$，其中：$n_1 \geq n_2 \geq \ldots \geq n_k \geq 1, k \geq 1$

正整数 n 的一个这种表示称为正整数 n 的一个划分。正整数 n 的不同的划分个数总和称为正整数 n 的划分数，记作 $p(n)$；另外，在正整数 n 的所有不同划分中，将最大加数 n 不大于 m 的划分个数记作 $q(n, m)$，则当 $n = 10$ 时，$p(n) =$ ________。

A、$q(8, 8)$ B、$1 + q(9, 9)$ C、$2 + q(10, 8)$ D、$q(10, 8)$

36、某一类算法是从问题某一初始或推测值出发，每一步尽可能去求得最佳解，当达到某一步不能继续时终止，使用每一步骤的最佳解来合成最后的解答，此类算法通称为_______。

A、穷举法 B、倒推法 C、贪心法 D、回溯法

37、n 个人拎着水桶在一个水龙头前面排队打水，水桶有大有小，水桶必须打满水，水流恒定，如下________说法不正确。

A、让水桶大的人先打水，可以使得每个人排队时间之和最小

B、让水桶小的人先打水，可以使得每个人排队时间之和最小

C、让水桶小的人先打水，同一时间内可以让尽可能多的人打上水

D、要在尽可能短的时间内 n 个人都打完水，按照什么顺序都一样

38、有 n 个独立的作业 1 , 2 , .. , n，由 m 台相同的机器加工处理。作业 i 所需的处理时间为 t_i，约定任何作业可以在任何一台机器上加工处理，但未完工前不允许中断，任何作业不能拆分成更小的作业。多机调度问题要求给出一种作业调度方案，使所给的 n 个作业在尽可能短的时间内由 m 台机器加工处理完成($n > m$)，对于多级调度问题，比较合适的贪心策略是________。

A、作业从小到大依次分配给空闲的机器

B、作业从大到小依次分配给空闲的机器

C、每个机器分配一样的作业数

D、每个作业随机地分配给一台空闲机器

39、下列问题中，________不能使用贪心法解决。

A、单源最短路径问题 B、n 皇后问题

C、最小生成树问题 D、背包问题

40、________是贪心算法与动态规划算法的共同点。

A、重叠子问题 B、构造最优解

C、贪心选择性质 D、最优子结构性质

41、贪心算法与动态规划算法的主要区别是________。

A、最优子结构 B、贪心选择性质

C、构造最优解 D、定义最优解

42、下面特征中，________是贪心算法的基本要素。
A、重叠子问题　　B、构造最优解
C、贪心选择性质　　D、定义最优解

43、能采用贪心算法求最优解的问题，一般具有的重要性质为________。
A、最优子结构性质与贪心选择性质
B、重叠子问题性质与贪心选择性质
C、最优子结构性质与重叠子问题性质　　D、预排序与递归调用

44、*Huffman* 编码的贪心算法所需的计算时间为________。
A、$O(n2^n)$　　B、$O(n\log n)$　　C、$O(2^n)$　　D、$O(n)$

45、图书管理系统对图书管理是按图书的序号从小到大进行管理的，如果要查找一本已知序号的书，则能快速查找的算法是_______。
A、枚举算法　　B、解析算法　　C、折半查找　　D、快速排序

46、二分查找算法是利用________实现的算法。
A、分治策略　　B、动态规划法　　C、贪心法　　D、回溯法

47、使用二分查找算法在 1000 个有序元素表中查找一个特定元素，在最坏情况下，搜索总共需要比较的次数为________。
A、10　　B、11　　C、500　　D、1000

48、在寻找 n 个元素中第 k 小元素问题中，如快速排序算法思想，运用分治算法对 n 个元素进行划分，对于选择划分基准，下面________的解释最合理。
A、随机选择一个元素作为划分基准
B、取子序列的第一个元素作为划分基准
C、用中位数的中位数方法寻找划分基准
D、以上皆可行，但不同方法，算法复杂度上界可能不同

49、在棋盘覆盖问题中，对于 $2^k \times 2^k$ 的特殊棋盘(有一个特殊方块)，所需的 L 型骨牌的个数是________。
A、$(4^k - 1) / 3$　　B、$2^k / 3$　　C、4^k　　D、2^k

50、以下问题中，不能使用分治法求解的是________。
A、棋盘覆盖问题　B、选择问题　C、归并排序　D、0-1 背包问题

51、实现循环赛日程表利用的算法是________。
A、分治策略　　B、动态规划法　　C、贪心法　　D、回溯法

52、*Strassen* 矩阵乘法是利用________实现的算法。
A、分治策略　　B、动态规划法　　C、贪心法　　D、回溯法

53、使用分治法求解不需要满足的条件是________。
A、子问题必须是一样的　　B、子问题不能重复
C、子问题的解可以合并　　D、原问题和子问题使用相同的方法求解

54、实现大整数的乘法是利用的算法________。
A、贪心法　　B、动态规划法　　C、分治策略　　D、回溯法

55、实现棋盘覆盖算法利用的算法是________。
A、分治法　　B、动态规划法　　C、贪心法　　D、回溯法

56、实现归并排序利用的算法是________。
A、分治策略　　B、动态规划法　　C、贪心法　　D、回溯法

57、下列过程中，不是动态规划算法基本步骤的是________。
A、找出最优解的性质　　B、构造最优解
C、算出最优解　　D、定义最优解

58、下列要素中，________是动态规划算法基本要素。
A、定义最优解　　B、构造最优解
C、算出最优解　　D、子问题重叠性质

59、实现最大子段和利用的算法是________。
A、分治策略　　B、动态规划法　　C、贪心法　　D、回溯法

60、一个问题可用动态规划算法或贪心算法求解的关键特征是问题的________。
A、重叠子问题　　B、最优子结构性质
C、贪心选择性质　　D、定义最优解

61、动态规划算法的基本要素为________。
A、最优子结构性质与贪心选择性质
B、重叠子问题性质与贪心选择性质
C、最优子结构性质与重叠子问题性质
D、预排序与递归调用

62、下列陈述中，不是动态规划算法基本步骤的是________。
A、找出最优解的性质　　B、构造最优解
C、算出最优解　　D、定义最优解

63、下列算法中，通常以自底向上的方式求得最优解的是________。
A、分治法　　B、动态规划法　　C、贪心法　　D、回溯法

64、矩阵连乘问题的算法可由________设计实现。
A、分支限界算法　　B、动态规划算法
C、贪心算法　　D、回溯算法

65、备忘录方法是________算法的变形。
A、分治法　　B、动态规划法　　C、贪心法　　D、回溯法

66、给出一个由 n 个数组成的序列 $A[1..n]$，要求找出它的最长单调上升子序列，设 $m[i](1 \le i \le n)$，表示以 $A[i]$ 结尾的最长单调上升子序列的长度，则 $m[1] = 1$，$m[i](1 < i \le n)$为________。
A、$m[i] = 1 + \mathbf{max}\{0, m[k](A[k] < A[i], 1 \le k < i)\}$
B、$m[i] = 1 + m[k]\ (k = i - 1 \ \&\&\ i > 1)$
C、$m[i] = 1 + \mathbf{max}\{0, m[k]\ (A[k] \le A[i], 1 \le k < i)\}$
D、$m[i] = \mathbf{max}\{0, m[k](A[k] < A[i], 1 \le k < i)\}$

67、对于矩阵连乘问题，其递推关系式为：

$$m[i,j] = \begin{cases} 0 & i = j \\ \min\limits_{i \le k < j}\{m[i,k] + m[k+1,j] + p_{i-1}p_k p_j\} & i < j \end{cases}$$

其中 $m[i, j]$ 为计算 $A[i, j]$ 所需的最少数乘次数，现在要计算矩阵连乘积 $A_1A_2A_3A_4$，其中各矩阵维数分别为：

A_1	A_2	A_3	A_4
50×10	10×40	40×30	30×5

则计算矩阵连乘积 $A_1A_2A_3A_4$ 所需要的最少数乘次数为________。
A、10500　　B、8000　　C、8500　　D、14500

68、最长公共子序列算法利用的算法是________。
A、分支限界法　　B、动态规划法　　C、贪心法　　D、回溯法

69、下列算法中，通常以深度优先方式系统搜索问题解的是________。
A、备忘录法　　B、动态规划法　　C、贪心法　　D、回溯法

70、回溯法搜索状态空间树是按照________的顺序。

A、中序遍历　　B、广度优先遍历

C、深度优先遍历　　D、层次优先遍历

71、某一类算法是在搜索尝试中找问题的解，当不满足求解条件时就退回，再尝试别的路径，此类算法通称为_______。

A、倒推法　　B、分治法　　C、贪心法　　D、回溯法

72、下面的________函数，是回溯法中为避免无效搜索采取的策略。

A、递归　　B、剪枝　　C、随机数　　D、搜索

73、回溯法的效率不依赖于以下________因素。

A、满足显约束的值的个数　　B、计算约束函数的时间

C、计算限界函数的时间　　D、确定解空间的时间

74、以深度优先方式系统搜索问题解的算法称为________。

A、分支限界算法　　B、概率算法

C、贪心算法　　D、回溯算法

75、程序块________是回溯法中遍历排列树的算法框架程序。

A、
```
void backtrack(int t)
{
    if (t > n)
        output(x);
    else
        for (int i = t; i <= n; i ++)
        {
            swap(x[t], x[i]);
            if (legal(t))
                backtrack(t + 1);
            swap(x[t], x[i]);
        }
}
```

B、
```
void backtrack(int t)
{
    if (t > n)
        output(x);
    else
        for (int i = 0; i <= 1; i ++)
        {
            x[t] = i;
            if (legal(t))
                backtrack(t + 1);
        }
}
```

C、
```
void backtrack(int t)
{
    if (t > n)
        output(x);
    else
        for (int i = 0; i <= 1; i ++)
        {
            x[t] = i;
            if (legal(t))
                backtrack(t – 1);
        }
}
```

D、
```
void backtrack(int t)
{
    if (t > n)
        output(x);
    else
        for (int i = t; i <= n; i ++)
        {
            swap(x[t], x[i]);
            if (legal(t))
                backtrack(t + 1);
        }
}
```

76、最大效益优先是________的一种搜索方式。

A、分支限界法　　B、动态规划法　　C、贪心法　　D、回溯法

77、优先队列通常用以下________数据结构来实现。

A、栈　　B、堆　　C、队列　　D、二叉查找树

78、关于回溯搜索法的介绍，下面________是不正确描述。

A、回溯法有“通用解题法”之称，它可以系统地搜索一个问题的所有解或任意解

B、回溯法是一种既带系统性又带有跳跃性的搜索算法

C、回溯算法在生成解空间任一结点时，先判断该结点是否可能包含问题的解，如果不包含，则跳过对该结点为根的子树搜索，逐层向祖先结点回溯

D、回溯算法需要借助队列这种结构来保存从根结点到当前扩展结点的路径

79、回溯法在问题的解空间树中，按________策略从根结点出发搜索解空间树。

A、广度优先　　B、活结点优先

C、扩展结点优先　　D、深度优先

80、回溯法解旅行售货员问题时的解空间树是________。

A、子集树　　B、排列树

C、深度优先生成树　　D、广度优先生成树

81、关于回溯算法和分支限界法，以下________是不正确描述。

A、回溯法中，每个活结点只有一次机会成为扩展结点

B、分支限界法中，活结点一旦成为扩展结点，就一次性产生其所有孩子结点，在这些孩子结点中，那些导致不可行解或者导致非最优解的孩子结点被舍弃，其余孩子加入活结点表中

C、回溯法采用深度优先的结点生成策略

D、分支限界法采用广度优先或者最大效益优先的结点生成策略

82、在分支限界算法中，根据从活结点表中选择下一扩展结点的不同方式可有几种常用分类，以下描述中，________最为准确。

A、采用 FIFO 队列的队列式分支限界法

B、采用最小值堆的优先队列式分支限界法

C、采用最大值堆的优先队列式分支限界法

D、三者都常用，针对具体问题可以选择采用其中某种更为合适的方式

83、以下对布线问题的描述中，________不正确。

A、布线问题的解空间是一个图

B、可以对方格阵列四周设置围墙，即增设标记的附加方格的预处理，使得算法简化对边界的判定

C、如果解一定存在，采用广度优先的标号法找到从起点到终点的布线方案不一定最短

D、采用先进先出的队列作为活结点表，以终点 *b* 为扩展结点或活结点队列为空作为算法结束条件

84、分支限界法与回溯法都是在问题的解空间树上搜索问题的解，二者_______。

A、求解目标不同，搜索方式相同

B、求解目标不同，搜索方式也不同

C、求解目标相同，搜索方式不同

D、求解目标相同，搜索方式也相同

85、某一类算法对有约束条件的最优化问题所有可行解有选择地搜索，将可行解空间不断地划分为越来越小的子集，并为每一个子集的解计算一个上界或者下界，此类算法通称为_______。

A、回溯法　　B、贪心法　　C、分支限界法　　D、动态规划

86、回溯法与分支限界法求解问题的解空间树不会是_______。

A、有序树　　B、子集树　　C、排列树　　　D、无序树

87、分支限界法解最大团问题时，活结点表的组织形式是________。

A、最小堆　　B、最大堆　　C、栈　　D、数组

88、对于含有 n 个元素的子集树问题，最坏情况下其解空间的叶子结点数目为________。

A、$n!$　　B、2^n　　C、$2^{n+1}-1$　　D、$\sum_{i=1}^{n}\frac{n!}{i!}$

89、下面方法中，不是分支限界法搜索方式的是________。

A、广度优先　　B、最小耗费优先

C、最大效益优先　　D、深度优先

90、优先队列式分支限界法选取扩展结点的原则是________。

A、先进先出　B、后进先出　C、结点优先级　　D、随机选取

91、广度优先是________的一种搜索方式。

A、分支限界法　　B、动态规划法　　C、贪心法　　D、回溯法

92、分支限界法在问题的解空间树中，按________策略，从根结点出发搜索解空间树。　　A、广度优先　B、活结点优先　　C、扩展结点优先　D、深度优先

93、常见的两种分支限界法为________。

A、广度优先分支限界法与深度优先分支限界法

B、队列式(FIFO)分支限界法与堆栈式分支限界法

C、排列树法与子集树法

D、队列式(FIFO)分支限界法与优先队列式分支限界法

94、分支限界法解旅行售货员问题时，活结点表的组织形式是________。

A、最小堆　　B、最大堆　　C、栈　　D、数组

95、采用广度优先策略搜索的算法是________。

A、分支限界法　　B、动态规划法　　C、贪心法　　D、回溯法

96、对于 0-1 背包问题和背包问题的解法，下面________陈述正确。

A、0-1 背包和背包问题都可用贪心算法求解

B、0-1 背包可以用贪心算法求解，但背包问题不能用贪心算法求解

C、0-1 背包不能用贪心算法求解，但可以使用动态规划或者搜索算法求解，而背包问题则可以用贪心算法求解

D、因为 0-1 背包不具有最优子结构性质，所以不能用贪心算法求解

97、0-1 背包问题的回溯算法所需的计算时间为________。

A、$O(n2^n)$　　B、$O(n\log n)$　　C、$O(2^n)$　　D、$O(n)$

98、下列算法中不能解决 0-1 背包问题的是________。

A、贪心法　　B、动态规划　　C、回溯法　　D、分支限界法

99、以下关于 0-1 背包问题的描述中，正确的是________。

A、可以使用贪心算法找到最优解

B、能找到多项式时间的有效算法

C、使用动态规划方法可求解任意 0-1 背包问题

D、对于同一背包与相同的物品，做背包问题取得的总价值一定大于等于做 0-1 背包问题

100、采用贪心算法的最优装载问题，主要计算量在于将集装箱依其重量从小到大排序，故算法的时间复杂度为________。

A、$O(n2^n)$　　B、$O(n\log n)$　　C、$O(2^n)$　　D、$O(n)$

101、背包问题的贪心算法所需的计算时间为________。

A、$O(n2^n)$　　B、$O(n\log n)$　　C、$O(2^n)$　　D、$O(n)$

102、求数列 1, 2, 3, …, n 的和，可以采用________算法。

A、解析法　　B、迭代法　　C、回溯法　　D、递归法

103、解决一个问题通常有多种方法，如果说一个算法“有效”是指________。

A、这个算法能在一定的时间和空间资源限制内将问题解决

B、这个算法能在人的反应时间内将问题解决

C、这个算法比其他已知算法都更快地将问题解决

D、这个算法能在较短的时间内编码实现

104、时间复杂性为多项式界的算法有________。

A、快速排序算法　　B、n 皇后问题

C、计算 π 值　　D、*Prim* 算法

105、已知 $f(1)=1$, $f(2)=3$，当 $n>2$ 时，$f(n)=2f(n-1)+3f(n-2)$，编程求 $f(100)$ 的值可以采用________算法。

A、穷举法　　B、递归法　　C、解析法　　D、贪心法

106、适用分治法所解决的问题一般具有________特征。

A、该问题的规模缩小到一定的程度就可以容易地解决

B、该问题可以分解为若干个规模较小的相同问题

C、利用该问题分解出的子问题的解可以合并为该问题的解

D、该问题所分解出的各个子问题是相互独立的

107、下列算法中，具有最优子结构的算法有________。

A、概率算法　B、贪心法　C、分支限界法　　D、动态规划法

108、下列问题中，________是典型的 NP 完全问题。

A、排序问题　　B、n 皇后问题

C、m-着色问题　　D、旅行商问题

109、适于递归实现的算法有________。

A、并行算法　　B、近似算法　　C、分治法　　D、回溯法

二、填空

1、算法的时间复杂性指算法中________的执行次数。

2、算法是由若干条指令组成的有穷序列，且要满足输入、________、确定性和________四条性质。

3、算法是指解决问题的________或者________。

4、设 D_n 表示大小为 n 的输入集合，$t(i)$ 表示输入为 i 时算法的运算时间，$p(i)$ 表示输入 i 出现的概率，则算法的平均情况下时间复杂性 $A(n)=$________。

5、算法的“确定性”指的是组成算法的每条________是清晰的，无歧义的。

6、计算一个算法时间复杂度通常可以计算________、________或者计算步。

7、已知一个分治算法耗费的计算时间 $T(n)$，$T(n)$ 满足如下递归方程：

$$T(n)=\begin{cases}O(1) & n<2\\ 2T(n/2)+O(n) & n\geq 2\end{cases}$$

解得此递归方可得 $T(n)=O($________$)$。

8、大整数乘积算法是用________来设计的。
9、快速排序算法是基于________的一种排序算法。
10、出自于“平衡子问题”的思想，通常分治法在分割原问题，形成若干子问题时，这些子问题的规模都大致________。
11、使用二分搜索算法在 n 个有序元素表中搜索一个特定元素，在最佳情况下，搜索的时间复杂性为 O(________)，在最坏情况下，搜索的时间复杂性为 O(________)。
12、分治算法的时间复杂性常常满足如下形式的递归方程：

$$\begin{cases} f(n) = d & , n = n_0 \\ f(n) = af(n/c) + g(n) & , n > n_0 \end{cases}$$

其中的 $g(n)$ 表示________________________________。
13、分治算法的基本步骤包括________、________、________。
14、选择排序、插入排序和归并排序算法中，________算法是分治算法。
15、贪心算法中每次做出的贪心选择都是________最优选择。
16、贪心算法的基本要素是________性质和________性质。
17、________是贪心算法可行的第一个基本要素，也是贪心算法与动态规划算法的主要区别。
18、问题的________是该问题使用动态规划或者贪心法求解的关键特征。
19、所谓贪心选择性质是指________。
20、所谓最优子结构性质是指________。
21、矩阵连乘问题的算法可由________设计实现。
22、动态规划算法的基本思想是将待求解问题分解成若干________，先求解________，然后从这些________的解得到原问题的解。
23、动态规划和分治法在分解子问题方面的不同点是________。
24、动态规划算法有一个变形方法________。这种方法不同于动态规划算法“自底向上”的填充方向，而是“自顶向下”的递归方向，为每个解过的子问题建立了备忘录以备需要时查看，同样也可避免相同子问题的重复求解。
25、以深度优先方式系统搜索问题解的算法称为________。
26、回溯算法的基本思想是________。
27、回溯法搜索解空间树时，常用的两种剪枝函数为________和________。
28、回溯法是指________。
29、用回溯法解题的一个显著特征是在搜索过程中动态产生问题的解空间，在任何时刻，算法只保存从根结点到当前扩展结点的路径。如果解空间树中从根结点到叶子结点的最长路径的长度为 $h(n)$，则回溯法所需的计算空间通常为________。
30、回溯法的算法框架按照问题的解空间一般分为________算法框架与________算法框架。
31、用回溯法解 0-1 背包问题时，该问题的解空间结构为________结构。
32、用回溯法解批处理作业调度问题时，该问题的解空间结构为________结构。
33、使用回溯法进行状态空间树裁剪分支时一般有两个标准：约束条件和目标函数的界，n 皇后问题和 0-1 背包问题正好是两种不同的类型，其中同时使用约束条件和目标函数的界进行裁剪的是________，只使用约束条件进行裁剪的是________。

34、分支限界法主要有________分支限界法和________分支限界法。
35、优先队列式的分支限界法中，对于活结点表中的结点，其下界函数值越小，优先级越________。
36、分支限界法是一种既带有________又带有________的搜索算法。
37、以广度优先或以最小耗费方式搜索问题解的算法称为________。
38、旅行售货员问题的解空间树是________。
39、解决 0-1 背包问题可以使用动态规划、回溯法和分支限界法，其中不需要排序的是________，需要排序的是________、________。

三、应用题

1、按照渐近阶从低到高排序。

2^n 3^n $\log n$ $n!$ $n\log n$ n^2 n^n 10^3

2、用 O、W、Q 表示函数 f 与 g 之间阶次的关系，并分别指出下列所有函数中阶次最低和最高的函数：

1) $f(n) = 100$ $g(n) = \sqrt[100]{n}$
2) $f(n) = 6n + n\lceil\log n\rceil$ $g(n) = 3n$
3) $f(n) = n / \log n - 1$ $g(n) = 3^n$
4) $f(n) = 2^n + n^2$ $g(n) = 2\sqrt{n}$
5) $f(n) = \log_3 n$ $g(n) = \log_2 n$

3、某递归算法的时间复杂性 $T(n)$ 满足以下递归方程：

$$T(n) = \begin{cases} 1 & , n = 1 \\ T(\frac{n}{2}) + \log_2 n & , n > 1 \end{cases}$$

求 $T(n)$ 的渐近时间复杂度。
4、请将下面的递归调用函数改写为非递归函数：

```
void test(int &sum)
{
    int x;
    scanf("%d", &x);
    if (x == 0)
        sum = 0;
    else
    {
        test(sum);
        sum += x;
    }
    printf("%d ", sum);
}
```

5、求解如下递归式，已知 $T(1) = 1$。
1) $T(n) = T(n - 1) + n$
2) $T(n) = T(n / 2) + n$

6、n 个正整数组成的数列，进行如下操作：

每次擦去其中两个数 a 和 b，然后在数列中增加 1 个数 $a \times b + 1$，如此下去直至剩下一个数；

在所有按这种操作方式最后得到的数中，最大的记作 max，最小的记作 min，则该数列的极差定义为 $M = max - min$。

已知初始数列为 6, 7, 4, 2, 3, 5，求该数列的极差，并请写明求得级差的算法策略和完全过程。

7、真分数可以表示为埃及分数之和的形式，所谓埃及分数指分子为 1 的分数形式，例如：$\frac{7}{8} = \frac{1}{2} + \frac{1}{3} + \frac{1}{24}$，试用贪心法将分数$\frac{17}{39}$ 分解为埃及分数之和，并说明分解的详细过程。

8、某体育馆有一羽毛球场出租，现在总共有 10 位客户申请租用此羽毛球场，每个客户所租用的时间单元如下表所示，$s(i)$ 表示开始租用时刻，$f(i)$ 表示结束租用时刻，10 个客户的申请如下表所示：

i	1	2	3	4	5	6	7	8	9	10
$s(i)$	0	3	1	5	3	5	11	8	8	6
$f(i)$	6	5	4	9	8	7	13	12	11	10

同一时刻，该羽毛球场只能租借给一位客户，请设计一个租用安排方案，在这 10 位客户里面，使得体育馆能尽可能多地满足客户的需求，并依照上表的 10 个客户申请，计算出最多可以安排几位客户申请。

9、现有 8 位运动员要进行网球循环赛，设计一个满足以下要求的比赛日程表：

1) 每个选手必须与其他选手各赛一次；
2) 每个选手一天只能赛一次；
3) 循环赛一共进行 $n-1$ 天。

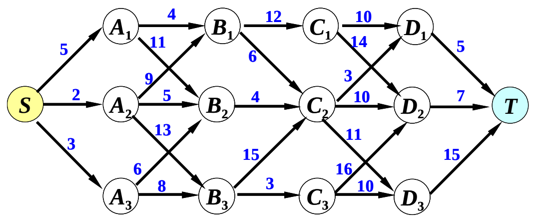
请利用分治法的思想，给这 8 位运动员设计一个合理的比赛日程。

10、有 4 个矩阵$\{A_1, A_2, A_3, A_4\}$，其中 A_i 与 A_{i+1} 可以相乘，$i = 1, 2, 3$，连乘积为 $A_1A_2A_3A_4$。

1) 在此四个矩阵连乘积问题中，不同子问题的个数总共有多少个？列举出所有的子问题。
2) 依此求得四个矩阵 $A_{10\times 30} \cdot B_{30\times 20} \cdot C_{20\times 10} \cdot D_{10\times 200}$ 运算量最小的乘积顺序，并将中间结果填入数组 $m[4][4]$ 中。

11、已知有向图如下，请用动态规划方法向后处理，求出从顶点 S 到顶点 T 的最长距离：

1) 写出详细的阶段划分；
2) 写出求最长距离的中间过程；
3) 求出最长距离，并说明最长距离时走过的路径。



12、某工业生产部门根据计划，拟将某种高效率的 5 台机器，分配给所属的 A、B、C 三个工厂，各工厂在获得这种机器后可以盈利，下表为 A、B、C 这三个工厂分别在拥有不同台数机器时的盈利值(万元)，求盈利最大的机器分配方案。

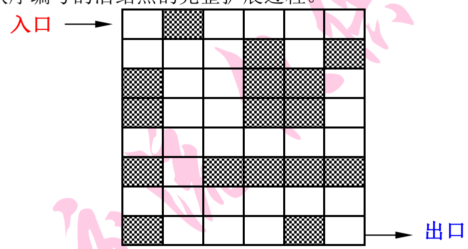
台数 / 工厂	0	1	2	3	4	5
A	0	3	7	9	12	13
B	0	5	10	11	11	11
C	0	4	6	11	12	12

13、对于给定的一个序列$(a_1, a_2, \ldots, a_n)$, $1 \le n \le 1000$，可以得到一些单调递增的子序列 $(a_{i1}, a_{i2}, \ldots, a_{ik})$，这里 $1 \le i_1 < i_2 < \ldots < i_k \le n$。

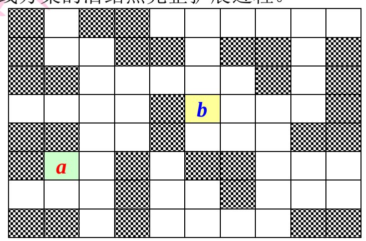
比如，序列(1, 7, 3, 5, 9, 4, 8)中，有一些上升子序列，如(1, 7), (3, 4, 8) 等。这些子序列中，最长的长度是 4，比如子序列(1, 3, 5, 8)。

对于给定的序列，求出最长上升子序列的长度，写出算法思想及递推式。

14、如下图所示迷宫，其中阴影方格为墙壁，空白的方格为路，如果分别采用深度优先搜索和广度优先搜索的方式在迷宫中寻找一条从入口到出口的路径，并约定深度优先搜索时尝试方向的次序为“右、下、左、上”，请在迷宫中标出按生成次序编号的活结点的完整扩展过程。



15、如下图所示印刷电路板将布线区域划分成 $n \times m$ 个方格，布线时，电路只能沿直线或者直角布线，为避免交叉，已经布线的方格做了封锁标记(图中的阴影部分)。请在印刷电路板中，标出用分支搜索法确定连接方格 a 中点到方格 b 中点的最短布线方案的活结点完整扩展过程。



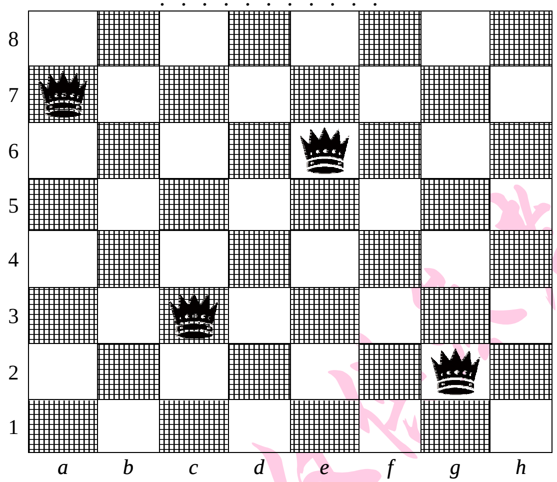
16、有这样一类特殊 0-1 背包问题，可选物品重量越轻的物品价值越高：

$n = 6$, $c = 20$, P = (4, 8, 15, 1, 6, 3), W = (5, 3, 2, 10, 4, 8),

其中 n 为物品个数，c 为背包载重量，P 表示物品的价值，W 表示物品的重量。对于此 0-1 背包问题，应如何选择放进去的物品，才能使到放进背包的物品总价值最大，能获得的最大总价值多少？

17、对于 4 皇后问题，画出用回溯法求解该问题时，在排列树上寻找所有解答的搜索过程。

18、下图所示国际象棋盘是某 8 皇后问题的解答中，部分皇后的安放位置示意，请根据这些皇后的位置求得其余 4 个皇后的位置。



四、问答

1、算法的复杂度分析涉及哪些方面？

2、什么是 P 类问题、NP 问题以及 NP 完全问题？描述这三者之间的关系，并列举几个常见的 NP 完全问题。

3、试述分治法的基本思想。

4、分治法所能解决的问题一般具有哪些特征？

5、将所给定序列 $a[1..n]$ 分为长度相等的两段 $a[1..n/2]$ 和 $a[n/2+1..n]$，分别求出这两段的最大子段和，则 $a[1..n]$ 的最大子段和有哪三种情形？

6、动态规划法的指导思想是什么？

7、设计动态规划算法有哪些主要步骤？

8、试比较分治法与动态规划法的异同。

9、写出回溯法搜索子集树的算法框架。

10、回溯法中常见哪两类典型的解空间树？各自的使用场合及时间复杂度？

11、分支限界法设计算法有哪些步骤？

12、常见的两种分支限界法的算法框架是什么？

13、分支限界法的搜索策略是什么？

14、试比较分支限界法与回溯法的异同。

五、算法设计

1、硬币找零

设有 $n(1 \le n \le 10)$ 种不同面值的硬币，各种硬币的面值存于数组 $T[1..n]$ 中。现要用这些面值的硬币来找零钱。各种面值可以使用的硬币个数存于数组 $Coins[1..n]$ 中，设计算法找出钱数 $L(0 \le L \le 20000)$ 的最少硬币个数。

2、最优服务次序问题

设有 n 个顾客同时等待一项服务，顾客 i 需要的服务时间为 t_i，$1 \le i \le n$，应该如何安排 n 个顾客的服务次序才能使平均等待时间达到最小？(平均等待时间是 n 个顾客等待服务时间的总和除以 n)。

3、查找第 k 最小元

给定线性表中有 n 个元素和一个整数 k，$1 \le k \le n$，找出这 n 个元素中第 k 小的元素，设计一个最坏时间复杂度为 $O(n)$ 的算法。

4、最长单调上升子序列

给定一个由 n 个数组成的序列，要求出该序列的最长单调上升子序列，请设计对应的算法并分析其时间复杂度，如果时间复杂度劣于 $O(n\log n)$ 的，将其优化为 $O(n\log n)$ 时间复杂度的算法。

参考答案

一、选择

1~30、	BACDA	CABBA	CCBDD	ACCDC	AAADC	CBCAD
31~60、	BDBCC	CAABD	BCABC	AADAD	AAACA	AADBB
61~90、	CABBB	AABDC	DADDA	ABDDA	ADCBC	DBBDC
91~101、	AADAA	CAADB	BB			

102~109、ABD　AC　ACD　BC　ABCD　BD　BCD　CD

二、填空

1、基本运算

2、输出、有穷性

3、一种方法、一个过程

4、$\sum_{i \in D_n} p(i)t(i)$

5、指令

6、循环次数、基本操作的频率

7、$n\log n$

8、分治法

9、分治策略

10、相等

11、1、$\log_2 n$

12、将规模为 n 的问题分解为子问题以及组合相应的子问题的解所需的时间

13、分解，递归，组合

14、归并排序算法

15、局部

16、贪心选择、最优子结构

17、贪心选择性质

18、最优子结构性质

19、所求问题的整体最优解可以通过一系列局部最优的选择，即贪心选择来达到

20、问题的最优解包含了其子问题的最优解

21、动态规划

22、子问题、子问题、子问题

23、前者分解出的子问题有重叠，而后者分解出的子问题相互独立（不重叠）

24、备忘录方法

25、回溯法

26、在问题的状态空间树上作带剪枝的 DFS 搜索（或者：DFS + 剪枝）

27、约束函数、限界函数

28、具有限界函数的深度优先生成法

29、$O(h(n))$

30、子集树、排列树

31、子集树

32、排列树

33、0-1 背包问题、n 皇后问题

34、队列式(FIFO)、优先队列式

35、高

36、系统性、跳跃性

37、分支限界法

38、排列树

39、动态规划、回溯法、分支限界法

三、应用题

1、

$10^3 \quad \log n \quad n\log n \quad n^2 \quad 2^n \quad 3^n \quad n! \quad n^n$

2、

1) $f(n) = O(g(n))$
2) $f(n) = \Omega(g(n))$
3) $f(n) = \Omega(g(n))$
4) $f(n) = O(g(n))$
5) $f(n) = \Theta(g(n))$

阶次最低的函数是 $f(n) = 100$，阶次最高的函数是 $g(n) = 3^n$

3、

设 $n = 2^k$，代入该递归方程得：

$$T(n) = T(\tfrac{n}{2}) + \log_2 n = T(\tfrac{n}{4}) + \log_2(\tfrac{n}{2}) + \log_2 n = \ldots = 1 + \sum_{i=0}^{k-1} \log_2 \frac{n}{2^i}$$

$$= 1 + k\log_2 n - \sum_{i=0}^{k-1} i = 1 + k\log_2 n - \frac{k(k-1)}{2} = \frac{1}{2}\log_2^2 n + \frac{1}{2}\log_2 n + 1$$

所以，$T(n) = \frac{1}{2}\log_2^2 n + \frac{1}{2}\log_2 n + 1 = \Theta(\log^2 n)$。

4、说明：该函数非尾递归，需要用栈保存中间输入的数据来再现原函数功能

```
void test1(int &sum)
{
    int x;
    stack<int> s;           //   辅助栈
    sum = 0;
    while (true)
    {
        scanf("%d", &x);
        s.push(x);          //   进栈
        if (x == 0)
            break;
    }
    while (! s.empty( ))
    {
        sum += s.top( );    //   取栈顶元素
        s.pop( );           //   出栈
        printf("%d ", sum);
    }
}
```

5、

1) $T(n) = T(n-1) + n = T(n-2) + n - 1 + n = \ldots$

$$= 1 + 2 + \ldots + n - 1 + n = \frac{n(n+1)}{2}$$

2) 设 $n = 2^k$，代入该递归方程得：

$$T(n) = T(\tfrac{n}{2}) + n = T(\tfrac{n}{4}) + \frac{n}{2} + n = T(\tfrac{n}{8}) + \frac{n}{4} + \frac{n}{2} + n = \ldots$$

$$= 1 + 2 + 4 + \ldots + 2^{k-1} + 2^k = 2 \times 2^k - 1 = 2n - 1$$

6、

1) 算法策略采用贪心法；
2) 将初始序列排序得到：7, 6, 5, 4, 3, 2；
3) 分别取出序列中最小值和最大值依次计算得到：

步骤	最大值	最小值
初始数据	7, 6, 5, 4, 3, 2	7, 6, 5, 4, 3, 2
第 1 步	7, 7, 6, 5, 4	43, 5, 4, 3, 2
第 2 步	21, 7, 7, 6,	216, 4, 3, 2
第 3 步	43, 21, 7	865, 3, 2
第 4 步	148, 43	2596, 2
第 5 步	6365	5193
结果	极差 $= max - min = 1172$	

7、详细步骤：

1) 用变量 A 表示分子 17，变量 B 表示分母 39
2) $C = INT(B / A) + 1 = INT(39 / 17) + 1 = 3$
3) 输出 1 / 3
4) $A_1 = A \times C - B = 17 \times 3 - 39 = 12$, $B_1 = B \times C = 3 \times 39 = 117$
5) $C_1 = INT(B_1 / A_1) + 1 = INT(117 / 12) + 1 = 10$
6) 输出 1 / 10
7) $A_2 = A_1 \times C_1 - B_1 = 12 \times 10 - 117 = 3$, $B_2 = B_1 \times C_1 = 117 \times 10 = 1170$
8) $A_2 > 1$，但 B_2 / A_2 为整数 390，输出 1 / 390，算法结束

于是得到埃及分数之和：$\frac{17}{39} = \frac{1}{3} + \frac{1}{10} + \frac{1}{390}$。

8、

将这 10 位客户的申请按照结束时间 $f(i)$ 递增排序，如下表：

i	1	2	3	4	5	6	7	8	9	10
$s(i)$	1	3	0	5	3	5	6	8	8	11
$f(i)$	4	5	6	7	8	9	10	11	12	13

1) 选择申请 1(1, 4)
2) 依次检查后续客户申请，只要与已选择的申请相容不冲突，则选择该申请，直到所有申请检查完毕：
   可以依次选择申请 4(5, 7)、申请 8(8, 11)、申请 10(11, 13)。

因此可以满足：申请 1(1, 4)、申请 4(5, 7)、申请 8(8, 11)、申请 10(11, 13)共 4 个客户申请，这已经是可以满足的最大客户人数。

9、

1	2	3	4	5	6	7	8
2	1	4	3	6	5	8	7
3	4	1	2	7	8	5	6
4	3	2	1	8	7	6	5
5	6	7	8	1	2	3	4

6	5	8	7	2	1	4	3	
7	8	5	6	3	4	1	2	
8	7	6	5	4	3	2	1	

10、

1) 不同子问题的个数为 4 + 3+ 2 + 1 = 10 个

A_1, A_2, A_3, A_4

A_1A_2, A_2A_3, A_3A_4

$A_1A_2A_3, A_2A_3A_4$

$A_1A_2A_3A_4$

2) 从矩阵个数从 2 个开始依次计算乘法的次数。

初始：$m_{11} = 0, m_{22} = 0, m_{33} = 0, m_{44} = 0$

2 个：$m_{12} = 10 \times 30 \times 20 = 6000$, $m_{23} = 30 \times 20 \times 10 = 6000$, $m_{34} = 20 \times 10 \times 200 = 40000$

3 个：$m_{13} = \mathbf{min}\{m_{12} + m_{33} + 10 \times 20 \times 10, m_{11} + m_{23} + 10 \times 30 \times 10\} = \mathbf{min}\{6000 + 0 + 2000, 0 + 6000 + 3000\} = 8000$

$m_{24} = \mathbf{min}\{m_{23} + m_{44} + 30 \times 10 \times 200, m_{22} + m_{34} + 30 \times 20 \times 200\} = \mathbf{min}\{6000 + 0 + 60000, 0 + 40000 + 120000\} = 66000$

4 个：$m_{14} = \mathbf{min}\{m_{11} + m_{24} + 10 \times 30 \times 200, m_{12} + m_{34} + 10 \times 20 \times 200, m_{13} + m_{44} + 10 \times 10 \times 200\} = \mathbf{min}\{0 + 66000 + 60000, 6000 + 40000 + 40000, 8000 + 0 + 20000\} = 28000$

相应地 m 数组为：$\begin{bmatrix} 0 & 60000 & 8000 & 28000 \\ & 0 & 6000 & 66000 \\ & & 0 & 40000 \\ & & & 0 \end{bmatrix}$

由计算过程可知各计算的中间过程：$m_{14} \Rightarrow m_{13} \cdot m_4$, $m_{13} \Rightarrow m_{12} \cdot m_3$

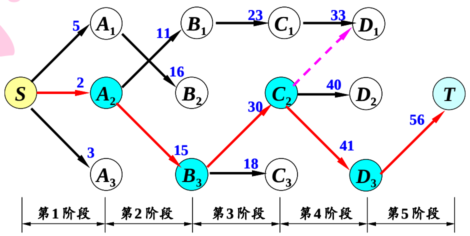
因此最少乘法次数的求积运算次序为：((A · B) · C) · D

11、

1) 最长距离 = 56

2) 方案为 $S \rightarrow A_2 \rightarrow B_3 \rightarrow C_2 \rightarrow D_3 \rightarrow T$

3) 阶段划分与中间结果如下图：



12、

第一步：单独考虑 A 项目时的最大收益

金 额	0	1	2	3	4	5
利润	0	3	7	9	12	13
方案	A: 0	A: 1	A: 2	A: 3	A: 4	A: 5

第二步：联合计算 A、B 项目时的最大收益

金 额	0	1	2	3	4	5
利润	0	5	10	13	17	19
方案	A: 0 B: 0	A: 0 B: 1	A: 0 B: 2	A: 1 B: 2	A: 2 B: 2	A: 3 B: 2

第三步：联合计算 A、B、C 项目时的最大收益

金 额	0	1	2	3	4	5
利润	0	5	10	14	17	21
方案	AB: 0 C: 0	AB: 1 C: 0	AB: 2 C: 0	AB: 2 C: 1	AB: 4 3 C: 0 1	AB: 4 2 C: 1 3

最后得到的最大收益是 21 万元，分配方案是：2 ⇒ A，2 ⇒ B，1 ⇒ C，或者 0 ⇒ A，2 ⇒ B，3 ⇒ C。

13、

设 $f(i)$ 表示：从左向右扫描过来直到以 $a[i]$ 元素结尾的序列，获得的最长上升子序列的长度，且子序列包含 $a[i]$元素($1 \le i \le n$)。

递推式为：$$f(i) = \begin{cases} 1 & i = 1 \\ \max\{f(j)+1 : a[i] > a[j]; 1 \le j < i\} & i > 1 \\ 1 & i > 1;\ 1 \le j < i, a[i] \le a[j] \end{cases}$$

即 $f(i)$ 是从 $f(1), f(2), \ldots$, 到 $f(i-1)$ 中找最大的一个值，加 1；或者就是 1。

依据元素 $a[i]$ 能否添加到前面已获得的最长上升子序列，如果能，就是前面已获得的最长上升子序列长度加一；如果不能，就将该一个元素作为一个单独子序列，长度为 1。

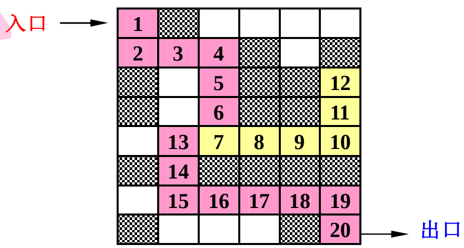
最后求得的整个序列最长公共子序列长度为 $\max\{f(i): 1 \le i \le n\}$。

例如对于序列：4 2 6 3 1 5 2

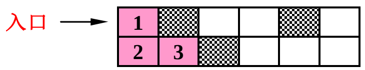
i	1	2	3	4	5	6	7
array	**4**	**2**	**6**	**3**	**1**	**5**	**2**
f(i)	**1**	**1**	**2**	**2**	**1**	**<u>3</u>**	**2**

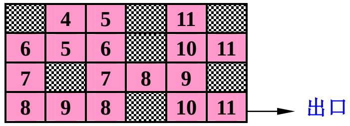
14、

深度优先

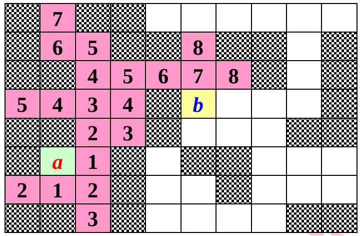


广度优先





15、

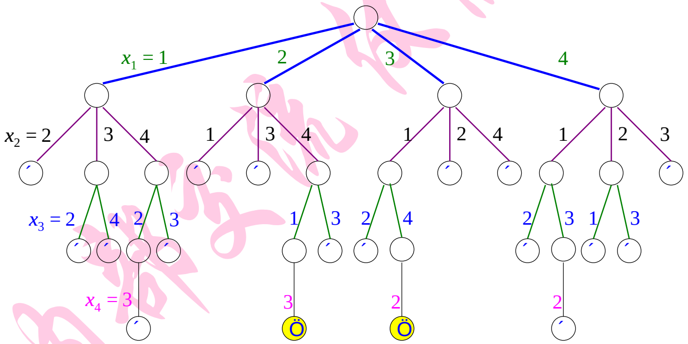


16、

因为该 0-1 背包问题比较特殊，恰好重量越轻的物品价值越高，所以优先取重量轻的物品放进背包。

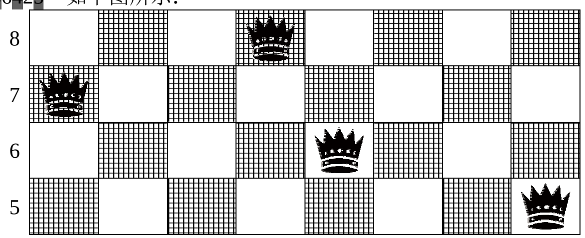
最终可以分别将重量为 2, 3, 4, 5 的三个物品放进背包，得到的价值和为 15 + 8 + 6 + 4 = 33，为最大值。

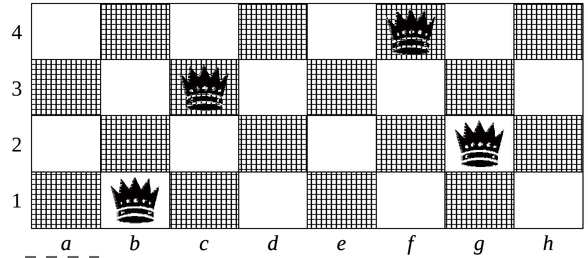
17、



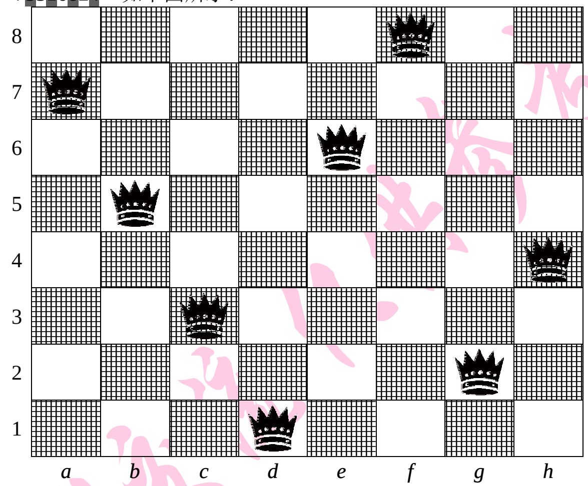
18、

71386425　如下图所示：





或者 75316824　如下图所示：



四、问答

1、

算法的复杂度分析涉及时间和空间两个方面。

2、

P 类问题是所有复杂度为多项式时间问题的集合；

所有的非确定性多项式时间可解的判定问题，称为 NP 类问题，虽然求解过程非确定性，但是可以确定性地在多项式时间内验证某个解是否正确；

某特殊的 NP 问题能通过一多项式时间算法转换为某个 NP 问题，该特殊 NP 问题称 NP 完全问题；

NP 问题包含了 P 类问题和 NP 完全问题；

常见的 NP 完全问题有推销员问题、0-1 背包问题等。

3、

分治法的基本思想是将一个规模为 n 的问题分解为 k 个规模较小的子问题，这些子问题互相独立且与原问题相同；

递归地解这些子问题，然后将各个子问题的解合并得到原问题的解。

4、

1) 该问题的规模缩小到一定的程度就可以容易地解决；

2) 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；
3) 利用该问题分解出的子问题的解可以合并为该问题的解；
4) 原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。

5、
1) $a[1 .. n]$ 的最大子段和与 $a[1 .. n / 2]$的最大子段和相同；
2) $a[1 .. n]$ 的最大子段和与 $a[n / 2 + 1 .. n]$ 的最大子段和相同；
3) $a[1 .. n]$ 的最大子段和为$\sum a_k(i \le k \le j)$，且 $1 \le i \le n / 2$, $n / 2 + 1 \le j \le n$。

6、
在做每一步决策时，列出各种可能的局部解，之后依据某种判定条件，舍弃那些 肯定不能得到最优解的局部解。这样在每一步都经过筛选，以每一步都是最优来保证全局最优。

7、
1) 找出最优解的性质，并刻划其结构特征；
2) 递归地定义最优值；
3) 以自底向上的方式计算出最优值；
4) 根据计算最优值时得到的信息，构造最优解。

8、
相同点：将待求解的问题分解成若干个子问题，先求解子问题，然后从这些子问题的解得到原问题的解。
不同点：适合于用动态规划法求解的问题，经分解得到的子问题往往不是互相独立的；而用分治法求解的问题，经分解得到的子问题往往是互相独立的。

9、
```
void backtrack (int t)
{
    if (t > n)
        output(x);
    else
        for (int i = 0; i <= 1; i ++)
        {
            x[t] = i;
            if (constraint(t) && bound(t))
                backtrack(t + 1);
        }
}
```

10、
回溯法中常见的两类典型解空间树是子集树和排列树。
当所给问题是从 n 个元素的集合 S 中找出满足某种性质的子集时，相应的解空间树称子集树，这类子集树通常有 2^n 个叶子结点，遍历子集树需 $O(2^n)$ 计算时间。
当所给问题是确定 n 个元素满足某种性质的排列时，相应的解空间树称排列树，这类排列树通常有 $n!$ 个叶子结点，遍历排列树需要 $O(n!)$ 计算时间。

11、
1) 针对所给问题，定义问题的解空间(对解进行编码)；
2) 确定易于搜索的解空间结构(按树或图组织解)；
3) 以广度优先或以最小耗费(最大收益)优先的方式搜索解空间，并在搜索过程中用剪枝函数避免无效搜索。

12、
1) 队列式(FIFO)分支限界法：按照队列先进先出(FIFO)原则选取下一个节点为扩展节点。
2) 优先队列式分支限界法：按照优先队列中规定的优先级选取优先级最高的节点成为当前扩展节点。

13、
在扩展结点处，先生成其所有的儿子结点(分支)，然后再从当前的活结点表中选择下一个扩展结点。
为了有效地选择下一扩展结点，加速搜索的进程，在每一个活结点处，计算一个函数值(限界)，并根据函数值，从当前活结点表中选择一个最有利的结点作为扩展结点，使搜索朝着解空间上有最优解的分支推进，以便尽快地找出一个最优解。

14、
相同点：都是一种在问题的解空间树 T 中搜索问题解的算法。
不同点：1) 求解目标不同；2) 搜索方式不同；3) 对扩展结点的扩展方式不同；4) 存储空间的要求不同。

五、算法设计的求解思路

1、
使用贪心策略：先尽量多地找最大面值，然后时第二大面值…。

2、
使用贪心策略：最短服务时间优先。
将 n 个顾客的服务时间 t_i 按照由小到大排序，n 个顾客的服务调度方案即为排序后的顺序，即可使得平均等待时间最小。

3、
使用快速排序一趟划分算法：
1) 如果划分后基准值的位置为 k，则基准值就是第 k 小元素；
2) 如果基准值的位置小于 k，则继续划分基准值的前半序列；
3) 如果基准值的位置大于 k，则继续划分基准值的后半序列。

4、
1) 从 $a[n]$ 开始查找时，不降序列长度只能为 1；
2) 如果从 $a[n – 1]$ 开始查找，则：
如果 $a[n – 1] < a[n]$，则序列长度为 2；
如果 $a[n – 1] > a[n]$，则序列长度仅为 1。
3) 如果从 $a[i]$ 开始，此时最长不下降序列：
在 $a[i + 1], a[i + 2], …, a[n]$ 中，找出一个比 $a[i]$ 大且最长的不下降序列，作为它的后继。

5、补充
参见教材中的枚举法、迭代法、倒推法、贪心法、分治法以及动态规划中的经典或者简单算法；排列和组合生成算法。