一。选择题

1、二分搜索算法是利用（ A ）实现的算法。

A、分治策略 B、动态规划法 C、贪心法 D、回溯法

2、下列不是动态规划算法基本步骤的是（ A ）。

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

3、最大效益优先是（ A ）的一搜索方式。

A、分支界限法 B、动态规划法 C、贪心法 D、回溯法

4、在下列算法中有时找不到问题解的是（ B ）。

A、蒙特卡罗算法 B、拉斯维加斯算法 C、舍伍德算法 D、数值概率算法

5. 回溯法解旅行售货员问题时的解空间树是（ B ）。

A、子集树 B、排列树 C、深度优先生成树 D、广度优先生成树

6. 下列算法中通常以自底向上的方式求解最优解的是（ B ）。

A、备忘录法 B、动态规划法 C、贪心法 D、回溯法

7、衡量一个算法好坏的标准是（C ）。

A 运行速度快 B 占用空间少 C 时间复杂度低 D 代码短

8、以下不可以使用分治法求解的是（D ）。

A 棋盘覆盖问题 B 选择问题 C 归并排序 D 0/1 背包问题

9. 实现循环赛日程表利用的算法是（ A ）。

A、分治策略 B、动态规划法 C、贪心法 D、回溯法

10、下列随机算法中运行时有时候成功有时候失败的是（C ）

A 数值概率算法 B 舍伍德算法 C 拉斯维加斯算法 D 蒙特卡罗算法

11. 下面不是分支界限法搜索方式的是（ D ）。

A、广度优先 B、最小耗费优先 C、最大效益优先 D、深度优先

12. 下列算法中通常以深度优先方式系统搜索问题解的是（ D ）。

A、备忘录法 B、动态规划法 C、贪心法 D、回溯法

13.备忘录方法是那种算法的变形。（ B ）

A、分治法　　B、动态规划法　　C、贪心法　　D、回溯法

14．哈弗曼编码的贪心算法所需的计算时间为（　　B　　）。

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

15．分支限界法解最大团问题时，活结点表的组织形式是（　　B　　）。

A、最小堆　　B、最大堆　　C、栈　　D、数组

16．最长公共子序列算法利用的算法是（　　B　　）。

A、分支界限法　　B、动态规划法　　C、贪心法　　D、回溯法

17．实现棋盘覆盖算法利用的算法是（　　A　　）。

A、分治法　　B、动态规划法　　C、贪心法　　D、回溯法

18.下面是贪心算法的基本要素的是（　　C　　）。

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

19.回溯法的效率不依赖于下列哪些因素（　D　）

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

20.下面哪种函数是回溯法中为避免无效搜索采取的策略（　　B　　）

A. 递归函数　　B.剪枝函数　　C。随机数函数　　D.搜索函数

21、下面关于 NP 问题说法正确的是（B ）

A NP 问题都是不可能解决的问题

B P 类问题包含在 NP 类问题中

C NP 完全问题是 P 类问题的子集

D NP 类问题包含在 P 类问题中

22、蒙特卡罗算法是（　　B　　）的一种。

A、分支界限算法　　B、概率算法　　C、贪心算法　　D、回溯算法

23.下列哪一种算法不是随机化算法（　　C　　）

A. 蒙特卡罗算法 B. 拉斯维加斯算法 C.动态规划算法 D.舍伍德算法

24.（　　D　　）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

25. 矩阵连乘问题的算法可由（　　B）设计实现。

A、分支界限算法　　B、动态规划算法　　C、贪心算法　　D、回溯算法

26. 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　　）。

A、最小堆　　B、最大堆　　C、栈　　D、数组

27、Strassen 矩阵乘法是利用（　　A　　）实现的算法。

A、分治策略　　B、动态规划法　　C、贪心法　　D、回溯法

29、使用分治法求解不需要满足的条件是（A ）。

A 子问题必须是一样的

B 子问题不能够重复

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解

30、下面问题（B ）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 最小花费生成树问题　　D 背包问题

31、下列算法中不能解决 0/1 背包问题的是（A ）

A 贪心法 B 动态规划 C 回溯法 D 分支限界法

32、回溯法搜索状态空间树是按照（C ）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

33、下列随机算法中运行时有时候成功有时候失败的是（C ）

A 数值概率算法 B 舍伍德算法 C 拉斯维加斯算法 D 蒙特卡罗算法

34. 实现合并排序利用的算法是（　　A　　）。

A、分治策略　　B、动态规划法　　C、贪心法　　D、回溯法

35. 下列是动态规划算法基本要素的是（　　D　　）。

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

36. 下列算法中通常以自底向下的方式求解最优解的是（　　B　　）。

A、分治法　　B、动态规划法　　C、贪心法　　D、回溯法

37. 采用广度优先策略搜索的算法是（　　A　　）。

A、分支界限法　　B、动态规划法　　C、贪心法　　D、回溯法

38、合并排序算法是利用（　　A　　）实现的算法。

A、分治策略　　B、动态规划法　　C、贪心法　　D、回溯法

39、在下列算法中得到的解未必正确的是（　　B　　）。

A、蒙特卡罗算法　　B、拉斯维加斯算法　　C、舍伍德算法　　D、数值概率算法

40、背包问题的贪心算法所需的计算时间为（　　B　　）

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

41．实现大整数的乘法是利用的算法（　　C　　）。

A、贪心法　　B、动态规划法　　C、分治策略　　D、回溯法

42．0-1 背包问题的回溯算法所需的计算时间为（　　A　　）

A、O（n2n）　　B、O（nlogn）　　C、O（2n）　　D、O（n）

43．采用最大效益优先搜索方式的算法是（　　A　　）。

A、分支界限法　　B、动态规划法　　C、贪心法　　D、回溯法

44．贪心算法与动态规划算法的主要区别是（　　B　　）。

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

45．实现最大子段和利用的算法是（　　B　　）。

A、分治策略　　B、动态规划法　　C、贪心法　　D、回溯法

46.优先队列式分支限界法选取扩展结点的原则是（　　C　　）。

A、先进先出　　B、后进先出　　C、结点的优先级　　D、随机

47.背包问题的贪心算法所需的计算时间为（　　B　　）。

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

48、广度优先是（　　A　　）的一搜索方式。

A、分支界限法　　B、动态规划法　　C、贪心法　　D、回溯法

49、舍伍德算法是（　　B　　）的一种。

A、分支界限算法　　B、概率算法　　C、贪心算法　　D、回溯算法

50、在下列算法中有时找不到问题解的是（　　B　　）。

A、蒙特卡罗算法　　B、拉斯维加斯算法　　C、舍伍德算法　　D、数值概率算法

51 下列哪一种算法是随机化算法（　　D　　）

A. 贪心算法 B. 回溯法 C.动态规划算法 D.舍伍德算法

52. 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（　　B　　）。

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

53.采用贪心算法的最优装载问题的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　（　　B　　）。

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

54. 以深度优先方式系统搜索问题解的算法称为　（　　D　）　　。

A、分支界限算法　　B、概率算法　　C、贪心算法　　D、回溯算法

55. 实现最长公共子序列利用的算法是（　　B　　）。

A、分治策略　　B、动态规划法　　C、贪心法　　D、回溯法

56、算法是由若干条指令组成的有穷序列，而且满足以下性质（ D　）

（1） 输入：有 0 个或多个输入

（2） 输出：至少有一个输出

（3） 确定性：指令清晰，无歧义

（4） 有限性：指令执行次数有限，而且执行时间有限

A (1)(2)(3)　　B(1)(2)(4)　　C(1)(3)(4)　　D (1) (2)(3)(4)

57、函数 $32^n+10n\log^n$ 的渐进表达式是(B　).

A. 2^n　　B. 32^n　　C. $n\log^n$　　D. $10n\log^n$

58、大整数乘法算法是(　A　).算法

A.分治　B.贪心 C.动态规划　D.穷举

59、用动态规划算法解决最大字段和问题，其时间复杂性为(B　).

A.logn　　B.n　　C.n^2　　D.nlogn

60、解决活动安排问题，最好用（B　）算法

A.分治　B.贪心 C.动态规划　D.穷举

61、设 f(N),g(N)是定义在正数集上的正函数,如果存在正的常数 C 和自然数 N_0,使

得当 $N \geq N_0$ 时有 $f(N) \leq Cg(N)$,则称函数 f(N)当 N 充分大时有下界 g(N),记作

$f(N) \in O(g(N))$,即 f(N)的阶(A)g(N)的阶.

A.不高于 B.不低于 C.等价于 D.逼近

62、回溯法在解空间树 T 上的搜索方式是(A).

A.深度优先 B.广度优先 C.最小耗费优先 D.活结点优先

63、回溯算法和分支限界法的问题的解空间树不会是(D).

A.有序树 B.子集树 C.排列树 D.无序树

64、在对问题的解空间树进行搜索的方法中,一个活结点最多有一次机会成为活结点的是(B).

A.回溯法 B.分支限界法 C.回溯法和分支限界法 D.回溯法求解子集树问题

65、从活结点表中选择下一个扩展结点的不同方式将导致不同的分支限界法,以下除(C)之外都是最常见的方式.

A.队列式分支限界法 B.优先队列式分支限界法

C.栈式分支限界法 D.FIFO 分支限界法

二、 填空题

1. 算法的复杂性有 时间 复杂性和 空间 复杂性之分。

2、程序是 算法 用某种程序设计语言的具体实现。

3、算法的“确定性”指的是组成算法的每条 指令 是清晰的，无歧义的。

4. 矩阵连乘问题的算法可由 动态规划 设计实现。

5、拉斯维加斯算法找到的解一定是 正确解。

6、算法是指解决问题的 一种方法 或 一个过程 。

7、从分治法的一般设计模式可以看出，用它设计出的程序一般是 递归算法 。

8、问题的 最优子结构性质 是该问题可用动态规划算法或贪心算法求解的关键特征。

9、以深度优先方式系统搜索问题解的算法称为 回溯法 。

10、数值概率算法常用于 数值问题 的求解。

11、计算一个算法时间复杂度通常可以计算 循环次数 、 基本操作的频率 或计算步。

12、利用概率的性质计算近似值的随机算法是 数值概率算法，运行时以一定的概率得到正确解的随机算法是 蒙特卡罗算法 。

14、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 动态规划 ，需要排序的是 回溯法 ，分支限界法 。

15、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 0/1 背包问题 ，只使用约束条件进行裁剪的是 N 皇后问题 。

16、 贪心选择性质 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。

17、矩阵连乘问题的算法可由 动态规划 设计实现。

18、拉斯维加斯算法找到的解一定是 正确解。

19. 贪心算法的基本要素是 贪心选择 质和 最优子结构 性质 。

21. 动态规划算法的基本思想是将待求解问题分解成若干 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。

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

23、大整数乘积算法是用 分治法 来设计的。

24、以广度优先或以最小耗费方式搜索问题解的算法称为 分支限界法 。

25、舍伍德算法总能求得问题的 一个解 。

26、 贪心选择性质 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。

27.快速排序算法是基于 分治策略 的一种排序算法。

28.动态规划算法的两个基本要素是. 最优子结构 性质和重叠子问题 性质 。

30.回溯法是一种既带有 系统性 又带有 跳跃性 的搜索算法。

31.分支限界法主要有 队列式（FIFO） 分支限界法和优先队列式 分支限界法。

32. 分支限界法是一种既带有 系统性 又带有 跳跃性 的搜索算法。

33. 回溯法搜索解空间树时，常用的两种剪枝函数为 约束函数 和 限界函数 。

34.任何可用计算机求解的问题所需的时间都与其 规模 有关。

35.快速排序算法的性能取决于 划分的对称性 。

36. Prim 算法利用 贪心 策略求解 最小生成树 问题，其时间复杂度是 $O(n^2)$ 。

37. 图的 m 着色问题可用 回溯 法求解，其解空间树中叶子结点个数是 m^n ，解空间树中每个内结点的孩子数是 m 。

三、算法填空

1.背包问题的贪心算法

```
void Knapsack(int n,float M,float v[],float w[],float x[])
{
        Sort(n,v,w);
        int i;
        for (i=1;i<=n;i++) x[i]=0;
        float c=M;
        for (i=1;i<=n;i++) {
           if (w[i]>c) break;
           x[i]=1;
           c - =w[i];
           }
```

```
        if (i<=n) x[i]=c/w[i];
}
```

2.最大子段和: 动态规划算法

```
int MaxSum(int n, int a[])
{
    int sum=0, b=0;     //sum 存储当前最大的 b[j], b 存储 b[j]
    for(int j=1;  j<=n;  j++)  {
        if (b>0)  b+= a[j] ;
        else  b=a[i];      ;     //一旦某个区段和为负，则从下一个位置累
和
          if(b>sum) sum=b;

     }
     return sum;
 }
```

3.快速排序

```
template<class Type>
void QuickSort (Type a[], int p, int r)
{
      if (p<r) {
        int q=Partition(a,p,r);
        QuickSort (a,p,q-1); //对左半段排序
        QuickSort (a,q+1,r); //对右半段排序
        }
}
```

4.排列问题

```
Template <class Type>
void perm(Type list[],  int k, int m )
```

```
{ //产生[list[k:m]的所有排列
    if(k==m)
     {  //只剩下一个元素
         for (int i=0;i<=m;i++)  cout<<list[i];
         cout<<endl;
    }
    else  //还有多个元素待排列，递归产生排列
       for (int i=k; i<=m; i++)
        {
           swap(list[k], list[i]);
           perm(list,k+1;m);
           swap(list[k],list[i]);
     }
  }
```

5. 给定已按升序排好序的 n 个元素 a[0:n-1]，现要在这 n 个元素中找出一特定元素 x。

据此容易设计出二分搜索算法：

```
template<class Type>
int BinarySearch(Type a[], const Type& x, int l, int r)
{
      while (l<=r ){
          int m = ((l+r)/2);
          if (x == a[m]) return m;
          if (x < a[m]) r = m-1; else l = m+1;
          }
    return -1;
}
```

6、合并排序描述如下：

```
template<class Type>
```

```
void Mergesort(Type a[ ], int left, int right)
{
if (left<right){
        int i=( left+right)/2;
        Mergesort(a, left, i );
        Mergesort(a, i+1, right);
        Merge(a,b, left, i, right);//合并到数组 b
        Copy(a,b, left, right ); //复制到数组 a
        }
}
```

7、以下是计算 x^m 的值的过程

```
int    power ( x, m )
{//计算 x^m 的值并返回。
        y=( 1 );i=m;
        While(i- - >0)
            y=y*x;
        (return y) ;
}
```

四、问答题

1.用计算机求解问题的步骤：

1、问题分析 2、数学模型建立 3、算法设计与选择 4、算法指标 5、算法分析 6、算法实现 7、程序调试 8、结果整理文档编制

2. 算法定义：

算法是指在解决问题时，按照某种机械步骤一定可以得到问题结果的处理过程

3.算法的三要素

1、操作 2、控制结构 3、数据结构

4. 算法具有以下 5 个属性：

有穷性：一个算法必须总是在执行有穷步之后结束，且每一步都在有穷时间内完成。

确定性：算法中每一条指令必须有确切的含义。不存在二义性。只有一个入口和一个出口

可行性：一个算法是可行的就是算法描述的操作是可以通过已经实现的基本运算执行有限次来实现的。

输入：一个算法有零个或多个输入，这些输入取自于某个特定对象的集合。

输出：一个算法有一个或多个输出，这些输出同输入有着某些特定关系的量。

5. 算法设计的质量指标：

正确性：算法应满足具体问题的需求；

可读性：算法应该好读，以有利于读者对程序的理解；

健壮性：算法应具有容错处理，当输入为非法数据时，算法应对其作出反应，而不是产生莫名其妙的输出结果。

效率与存储量需求：效率指的是算法执行的时间；存储量需求指算法执行过程中所需要的最大存储空间。一般这两者与问题的规模有关。

经常采用的算法主要有迭代法、分治法、贪婪法、动态规划法、回溯法、分支限界法

6. 迭代法:

也称“辗转法”，是一种不断用变量的旧值递推出新值的解决问题的方法。

7.利用迭代算法解决问题，需要做好以下三个方面的工作：

1)、确定迭代模型。在可以用迭代算法解决的问题中，至少存在一个直接或间接地不断由旧值递推出新值的变量，这个变量就是迭代变量。

2)、建立迭代关系式。所谓迭代关系式，指如何从变量的前一个值推出其下一个值的公式（或关系）。迭代关系式的建立是解决迭代问题的关键，通常可以使用递推或倒推的方法来完成。

3)、对迭代过程进行控制。在什么时候结束迭代过程？这是编写迭代程序必须考虑的问题。不能让迭代过程无休止地重复执行下去。迭代过程的控制通常可分为两种情况：一种是所需的迭代次数是个确定的值，可以计算出来；另一种是所需的迭代次数无法确定。对于前一种情况，可以构建一个固定次数的循环来实现对迭代过程的控制；对于后一种情况，需要进一步分析出用来结束迭代过程的条件。

8. 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。

9.分治法所能解决的问题一般具有以下几个特征：

（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）该问题所分解出的各个子问题是相互独立的，即子问题之间不包含公共的子子问题。

10、分治法的基本步骤

分治法在每一层递归上都有三个步骤：

（1）分解：将原问题分解为若干个规模较小，相互独立，与原问题形式相同的子问题；

（2）解决：若子问题规模较小而容易被解决则直接解，否则递归地解各个子问题；

（3）合并：将各个子问题的解合并为原问题的解。

11. 动态规划的基本思想

前文主要介绍了动态规划的一些理论依据，我们将前文所说的具有明显的阶段划分和状态转移方程的动态规划称为**标准动态规划**，这种标准动态规划是在研

究多阶段决策问题时推导出来的，具有严格的数学形式，适合用于理论上的分析。在实际应用中，许多问题的阶段划分并不明显，这时如果刻意地划分阶段法反而麻烦。一般来说，只要该问题可以划分成规模更小的子问题，并且原问题的最优解中包含了子问题的最优解（即满足最优子化原理），则可以考虑用动态规划解决。

动态规划的实质是**分治思想**和**解决冗余**，因此，**动态规划**是一种将问题实例分解为更小的、相似的子问题，并存储子问题的解而避免计算重复的子问题，以解决最优化问题的算法策略。

由此可知，动态规划法与分治法和贪心法类似，它们都是将问题实例归纳为更小的、相似的子问题，并通过求解子问题产生一个全局最优解。

贪心法的当前选择可能要依赖已经作出的所有选择，但不依赖于有待于做出的选择和子问题。因此贪心法自顶向下，一步一步地作出贪心选择；
而分治法中的各个子问题是独立的（即不包含公共的子问题），因此一旦递归地求出各子问题的解后，便可自下而上地将子问题的解合并成问题的解。

不足之处：如果当前选择可能要依赖子问题的解时，则难以通过局部的贪心策略达到全局最优解；如果各子问题是不独立的，则分治法要做许多不必要的工作，重复地解公共的子问题。

解决上述问题的办法是利用动态规划。该方法主要应用于最优化问题，这类问题会有多种可能的解，每个解都有一个值，而动态规划找出其中最优（最大或最小）值的解。若存在若干个取最优值的解的话，它只取其中的一个。在求解过程中，该方法也是通过求解局部子问题的解达到全局最优解，但与分治法和贪心法不同的是，动态规划允许这些子问题不独立，（亦即各子问题可包含公共的子问题）也允许其通过自身子问题的解作出选择，该方法对每一个子问题只解一次，并将结果保存起来，避免每次碰到时都要重复计算。

因此，动态规划法所针对的问题有一个显著的特征，即它所对应的子问题树中的子问题呈现大量的重复。动态规划法的关键就在于，对于重复出现的子问题，只在第一次遇到时加以求解，并把答案保存起来，让以后再遇到时直接引用，不必重新求解。

12、动态规划算法的基本步骤

设计一个标准的动态规划算法，通常可按以下几个步骤进行：

（1）划分阶段：按照问题的时间或空间特征，把问题分为若干个阶段。注意这若干个阶段一定要是有序的或者是可排序的（即无后向性），否则问题就无法用动态规划求解。

（2）选择状态：将问题发展到各个阶段时所处于的各种客观情况用不同的状态表示出来。当然，状态的选择要满足无后效性。

（3）确定决策并写出状态转移方程：之所以把这两步放在一起，是因为决策和状态转移有着天然的联系，状态转移就是根据上一阶段的状态和决策来导出本阶段的状态。所以，如果我们确定了决策，状态转移方程也就写出来了。但事实上，我们常常是反过来做，根据相邻两段的各状态之间的关系来确定决策。

（4）写出规划方程（包括边界条件）：动态规划的基本方程是规划方程的通用形式化表达式。

一般说来，只要阶段、状态、决策和状态转移确定了，这一步还是比较简单的。动态规划的主要难点在于理论上的设计，一旦设计完成，实现部分就会非常简单。根据动态规划的基本方程可以直接递归计算最优值，但是一般将其改为递推计算。实际应用当中经常不显式地按照上面步骤设计动态规划，而是按以下几个步骤进行：

（1）分析最优解的性质，并刻划其结构特征。

（2）递归地定义最优值。

（3）以自底向上的方式或自顶向下的记忆化方法（备忘录法）计算出最优值。

（4）根据计算最优值时得到的信息，构造一个最优解。

步骤（1）～（3）是动态规划算法的基本步骤。在只需要求出最优值的情形，步骤（4）可以省略，若需要求出问题的一个最优解，则必须执行步骤（4）。此时，在步骤（3）中计算最优值时，通常需记录更多的信息，以便在步骤（4）中，根据所记录的信息，快速地构造出一个最优解。

总结：动态规划实际上就是最优化的问题，是指将原问题的大实例等价于同一最优化问题的较小实例，自底向上的求解最小实例，并将所求解存放起来，存放的结果就是为了准备数据。与递归相比，递归是不断的调用子程序求解，是自

顶向下的调用和求解。

13. 分治法与动态规划法的相同点是：

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。

两者的不同点是：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。

14. 回溯法

回溯法也称为试探法，该方法首先暂时放弃关于问题规模大小的限制，并将问题的候选解按某种顺序逐一枚举和检验。当发现当前候选解不可能是解时，就选择下一个候选解；倘若当前候选解除了还不满足问题规模要求外，满足所有其他要求时，继续扩大当前候选解的规模，并继续试探。如果当前候选解满足包括问题规模在内的所有要求时，该候选解就是问题的一个解。在回溯法中，放弃当前候选解，寻找下一个候选解的过程称为回溯。扩大当前候选解的规模，以继续试探的过程称为向前试探。

15. 分支限界法：

这是一种用于求解组合优化问题的排除非解的搜索算法。类似于回溯法，分枝定界法在搜索解空间时，也经常使用树形结构来组织解空间。然而与回溯法不同的是，回溯算法使用深度优先方法搜索树结构，而分枝定界一般用宽度优先或最小耗费方法来搜索这些树。因此，可以很容易比较回溯法与分枝定界法的异同。相对而言，分枝定界算法的解空间比回溯法大得多，因此当内存容量有限时，回溯法成功的可能性更大。

算法思想：分枝限界（branch and bound）是另一种系统地搜索解空间的方法，它与回溯法的主要区别在于对 E-节点的扩充方式。每个活节点有且仅有一次机会变成 E-节点。当一个节点变为 E-节点时，则生成从该节点移动一步即可到达的所有新节点。在生成的节点中，抛弃那些不可能导出（最优）可行解的节点，其余节点加入活节点表，然后从表中选择一个节点作为下一个 E-节点。从活节点表中取出所选择的节点并进行扩充，直到找到解或活动表为空，扩充过程才结

束。

有两种常用的方法可用来选择下一个 E-节点（虽然也可能存在其他的方法）：

1) 先进先出（F I F O） 即从活节点表中取出节点的顺序与加入节点的顺序相同，因此活

节点表的性质与队列相同。

2) （优先队列）最小耗费或最大收益法在这种模式中，每个节点都有一个对应的耗费或收益。如果查找 一个具有最小耗费的解，则活节点表可用最小堆来建立，下一个 E-节点就是具有最小耗费 的活节点；如果希望搜索一个具有最大收益的解，则可用最大堆来构造活节点表，下一个 E-节点是具有最大收益的活节点

16. 分支限界法与回溯法的相同点是：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。

17. 分治法所能解决的问题一般具有的几个特征是：

（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。

18. 用分支限界法设计算法的步骤是：

(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。

19. 常见的两种分支限界法的算法框架：

（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。

20. 回溯法中常见的两类典型的解空间树是子集树和排列树。

当所给的问题是从 n 个元素的集合 S 中找出满足某种性质的子集时，相应的解空间树称为子集树。这类子集树通常有 2n 个叶结点，遍历子集树需 O(2n)计算时间 。

当所给的问题是确定 n 个元素满足某种性质的排列时，相应的解空间树称为排列树。这类排列树通常有 n!个叶结点。遍历排列树需要 O(n!)计算时间。

21. 分支限界法的搜索策略是：

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。

22. 请叙述动态规划算法与贪心算法的异同。

共同点：

都需要最优子结构性质，

都用来求有优化问题。

不同点：

动态规划：每一步作一个选择—依赖于子问题的解。

贪心方法：每一步作一个选择—不依赖于子问题的解。

动态规划方法的条件：子问题的重叠性质。

可用贪心方法的条件：最优子结构性质；贪心选择性质。

动态规划：自底向上求解；

贪心方法： 自顶向下求解。

可用贪心法时，动态规划方法可能不适用；

可用动态规划方法时，贪心法可能不适用。

23. 请说明动态规划方法为什么需要最优子结构性质。

答：

最优子结构性质是指大问题的最优解包含子问题的最优解。

动态规划方法是自底向上计算各个子问题的最优解，即先计算子问题的最优解，然后再利用子问题的最优解构造大问题的最优解，因此需要最优子结构.

24. 请说明：

(1)优先队列可用什么数据结构实现？

(2)优先队列插入算法基本思想？

(3)优先队列插入算法时间复杂度？

答：（1）堆。

（2）在小根堆中，将元素 x 插入到堆的末尾，

然后将元素 x 的关键字与其双亲的关键字比较，

若元素 x 的关键字小于其双亲的关键字，

则将元素 x 与其双亲交换，然后再将元素 x 与其新双亲的关键字相比，

直到元素 x 的关键字大于双亲的关键字，或元素 x 到根为止。

（3）*O(log n)*

25. 衡量算法时间效率的方法有哪两种？请叙述。

答：有事前分析法和事后分析法两种。

事后分析法：先将算法用程序设计语言实现，然后度量程序的运行时间。

事前分析法：算法的时间效率是问题规模的函数，假如，随着问题规模 n 的增长，算法执行时间的增长率和函数 f(n)的增长率相同，则可记作：

$$T(n)=〇(f(n))$$

称 T(n)为算法的渐进时间复杂度。简称时间复杂度。

26. 在算法复杂性分析中，O、Ω、Θ这三个记号的意义是什么？在忽略常数因子的情况下，O、Ω、Θ分别提供了算法运行时间的什么界？

答：

如果存在两个正常数 c 和 $N0$,对于所有的 $N \geq N0$，有$|f(N)| \leq C|g(N)|$，则记作：$f(N)=O(g(N))$。这时我们说 $f(N)$的阶不高于 $g(N)$的阶。

若存在两个正常数 C 和自然数 N0，使得当 $N \geq N0$ 时有$|f(N)| \geq C|g(N)|$，记为 $f(N)=\Omega(g(N))$。这时我们说 $f(N)$的阶不低于 $g(N)$的阶。

如果存在正常数 c1，c2 和 n0，对于所有的 n≥n0，有 c1|g(N)| ≤|f(N)| ≤ c2|g(N)|

则记作 $f(N)\Theta(g(N))$

O、Ω、Θ分别提供了算法运行时间的上界、下界、平均

27. 概率算法

很多算法的每一个计算步骤都是固定的，而概率算法允许算法在执行的过程中随机选择下一个计算步骤。许多情况下，当算法在执行过程中面临一个选择时，随机性选择常比最优选择省时。因此概率算法可在很大程度上降低算法的复杂度。

28.概率算法的一个基本特征

是对所求解问题的同一实例用同一概率算法求解两次可能得到完全不同的效果。这两次求解问题所需的时间甚至所得到的结果可能会有相当大的差别。

29. 概率算法大致分为四类:

数值概率算法，蒙特卡罗（Monte Carlo）算法，拉斯维加斯（Las Vegas）算法和舍伍德（Sherwood）算法。

30.数值概率算法

常用于数值问题的求解。这类算法所得到的往往是近似解。而且近似解的精度随计算时间的增加不断提高。在许多情况下，要计算出问题的精确解是不可能或没有必要的，因此用数值概率算法可得到相当满意的解。

31.蒙特卡罗算法

用于求问题的准确解。对于许多问题来说，近似解毫无意义。例如，一个判定问题其解为“是”或“否”，二者必居其一，不存在任何近似解答。又如，我们要求一个整数的因子时所给出的解答必须是准确的，一个整数的近似因子没有任何意义。用蒙特卡罗算法能求得问题的一个解，但这个解未必是正确的。求得正确解的概率依赖于算法所用的时间。算法所用的

时间越多，得到正确解的概率就越高。蒙特卡罗算法的主要缺点就在于此。一般情况下，无法有效判断得到的解是否肯定正确。

32.拉斯维加斯算法

不会得到不正确的解，一旦用拉斯维加斯算法找到一个解，那么这个解肯定是正确的。但是有时候用拉斯维加斯算法可能找不到解。与蒙特卡罗算法类似。拉斯维加斯算法得到正确解的概率随着它用的计算时间的增加而提高。对于所求解问题的任一实例，用同一拉斯维加斯算法反复对该实例求解足够多次，可使求解失效的概率任意小。

33.舍伍德算法

总能求得问题的一个解，且所求得的解总是正确的。当一个确定性算法在最坏情况下的计算复杂性与其在平均情况下的计算复杂性有较大差别时，可以在这个确定算法中引入随机性将它改造成一个舍伍德算法，消除或减少问题的好坏实例间的这种差别。舍伍德算法精髓不是避免算法的最坏情况行为，而是设法消除这种最坏行为与特定实例之间的关联性。

舍伍德算法 sherwood algorithm

舍伍德算法 一类概率算法的代称。此类算法总能给出所求问题的正确的解。... 当解决某一问题的确定性算法的平均情形复杂性比最坏情形复杂性低得多时，通过引入随机性来试图减少甚至消除“好”、“坏”实体之间这种时间上的差别，以期望较小的运行时间。例 ...

五、算法设计与分析题

1．用动态规划策略求解最长公共子序列问题：

（1）给出计算最优值的递归方程。

（2）给定两个序列 X={B,C,D,A}，Y={A,B,C,B}，请采用动态规划策略求出其最长公共子序列，要求给出过程。

答：

(1)

$$c[i,j]=\begin{cases}0 & \text{当}i=0\text{或}j=0\text{时}\\ c[i-1,j-1]+1 & \text{当}i,j>0\text{且}x_i=y_i\text{时}\\ \max(c[i,j-1],c[i-1,j]) & \text{当}i,j>0\text{且}x_i\neq y_i\text{时}\end{cases}$$

(2)

		Y	A	B	C	B
			0	0	0	0
X						
B	0	0	1	1	1	
C	0	0	1	2	2	
D	0	0	1	2	2	
A	0	1	1	2	2	

最长公共子序列：{ B C }

2. 对下列各组函数 f (n) 和 g (n)，确定 f (n) = O (g (n)) 或 f (n) = Ω (g (n))或 f(n) = θ (g(n))，并简要说明理由。

(1) $f(n)=2^n$； $g(n)=n!$

(2) $f(n)=\sqrt{n}$； $g(n)=\log n^2$

(3) f(n)=100； g(n)=log100

(4) $f(n)=n^3$； $g(n)=3^n$

(5) $f(n)=3^n$； $g(n)=2^n$

答：

(1) f(n) = O(g(n)) 因为 g(n)的阶比 f(n)的阶高。

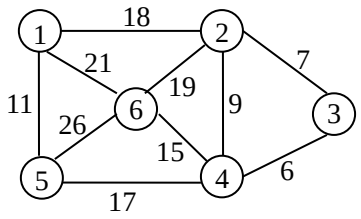
(2) f(n) = Ω (g(n)) 因为 g(n)的阶比 f(n)的阶低。

(3) f(n) = θ (g(n)) 因为 g(n)与 f(n)同阶。

(4) f(n) = O(g(n)) 因为 g(n)的阶比 f(n)的阶高。

(5) f(n) = Ω (g(n)) 因为 g(n)的阶比 f(n)的阶低。

3. 对下图所示的连通网络 G，用克鲁斯卡尔(Kruskal)算法求 G 的最小生成树 T, 请写出在算法执行过程中，依次加入 T 的边集 TE 中的边。说明该算法的贪心策略和算法的基本思想，并简要分析算法的时间复杂度。



答：

TE={(3,4), (2, 3), (1, 5), （4, 6）(4, 5）}

贪心策略是每次都在连接两个不同连通分量的边中选权值最小的边。

基本思想：首先将图中所有顶点都放到生成树中，然后每次都在连接两个不同连通分量的边中选权值最小的边，将其放入生成树中，直到生成树中有 n-1 条边。

时间复杂度为：O(eloge)

4. 请用分治策略设计递归的归并排序算法，并分析其时间复杂性（要求：分别给出 divide、conquer、combine 这三个阶段所花的时间，并在此基础上列出递归方程，最后用套用公式法求出其解的渐进阶）。

答 ： Template <class Type>

```
void MergeSort (Type a[ ], int left, int right)
{ if (left<right)
      { int i=（left+right）/2;
        MergeSort（a, left, i）;
        MergeSort（a, i+1, right）;
        Merge(a, b, left, right);
        Copy(a, b, left, right);
      }
}
```

Divide 阶段的时间复杂性： O(1)

Conquer 阶段的时间复杂性： 2T(n)

Combine 阶段的时间复杂性： Θ（n）

$$T(n)=\begin{cases}\theta(1) & \text{当} n=1\\ 2T(n/2)+\theta(n) & \text{当} n>1\end{cases}$$

用套用公式法：a=2, b=2, $n^{\log_b a} = n$, f(n)=n,　因为 f(n)与 $n^{\log_b a}$ 同阶，

∴T(n) = Θ（nlogn）

5、设有 $n=2^k$ 个运动员要进行循环赛,现设计一个满足以下要求的比赛日程表:

每个选手必须与其他 n-1 名选手比赛各一次;每个选手一天至多只能赛一次;循环赛要在最短时间内完成.

(1)（4 分）循环赛最少需要进行(n-1)天.

(2)（6 分）当 $n=2^3=8$ 时,请画出循环赛日程表:

	1	2	3	4	5	6	7
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

6、考虑用哈夫曼算法来找字符a,b,c,d,e,f 的最优编码。这些字符出现在文件中的频数之比为 20:10:6:4:44:16。要求：

（1）（4 分）简述使用哈夫曼算法构造最优编码的基本步骤；

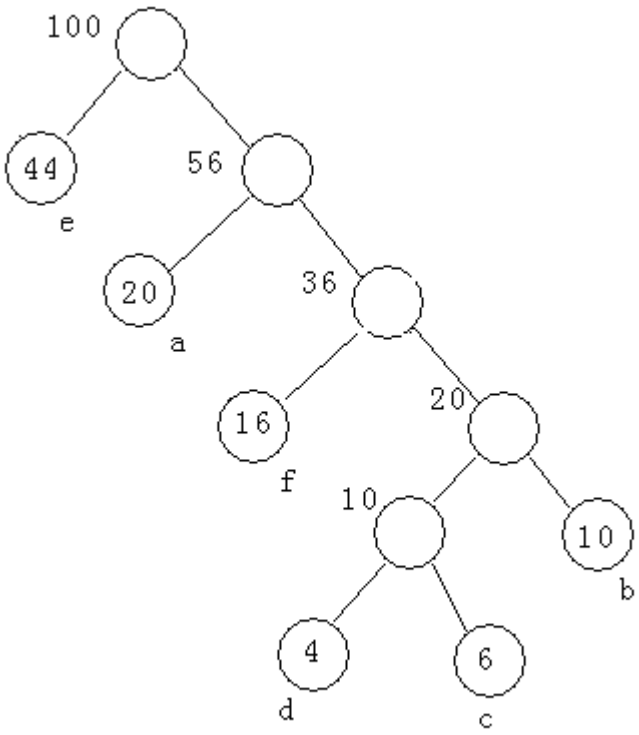
（2）（5 分）构造对应的哈夫曼树，并据此给出a,b,c,d,e,f 的一种最优编码。

解：1)、哈夫曼算法是构造最优编码树的贪心算法。其基本思想是，首先所有字符对应n 棵树构成的森林，每棵树只有一个结点，根权为对应字符的频率。然后，重复

下列过程n-1 次：将森林中的根权最小的两棵树进行合并产生一个新树，该新树根的两个子

树分别是参与合并的两棵子树，根权为两个子树根权之和。

2)、根据题中数据构造哈夫曼树如下图所示。



由此可以得出 a,b,c,d,e,f 的一组最优的编码：01,0000,00010,00011, 1,001。

7.考虑在序列A[1..n]中找最大最小元素的问题。一个分治算法描述如下：如果n≤2 就直接求解。否则，将序列等分成两个子序列A[1..n/2]和A[n/2+1..n]，分别找出这两子序列的最大最小元素x1,y1 和x2,y2；然后据此求出A[1..n]的最大元素x=max{x1,x2}及最小元素y=min{y1,y2}。请给出该算法计算时间T(n)满足的递归方程，并解方程来确定算法的时间复杂度。假定$n=2^k$（k 为正整数）。

答：

算法时间复杂度满足如下递归方程：

$T(n)=2T(n/2)+2$（$n>2$）；$T(2)=1$。

因为 $n=2^k$（k 为正整数），所以，

$T(n)= T(2^k)= 2T(2^{k-1})+2= 2^2T(2^{k-2})+ 2^2+2$

…

$= 2^{k-1}T(2)+ 2^{k-2}+\cdots+2^3+2^2+2$

$= 2^{k-1}+\cdots+2^3+2^2+2$。因此，$T(n)=\Theta(n)$。

8. 考虑使用动态规划方法求解下列问题：

01背包数据如下表，求：能够放入背包的最有价值的物品集合。

物品 i	重量 wi	价值 vi	承重量 W
1	w1=2	v1=12	W=5
2	w2=1	v2=10	
3	w3=3	v3=20	
4	w4=2	v4=15	

如设： V(i, j) —— 前 i 个物品中能够装入承重量 j 的背包中的最大总价值。请将如下递推式填写完整：

V(0, j) = 0（0个物品），V(i, 0) = 0（承重量0）

V(i, j) = V(i-1, j)　第 i 个物品不能装入，　j < wi　（超重）

V(i, j) = max {_____, _____ }　j > wi （不超重）

i在最优子集中　　i不在最优子集中

自底向上：按行或列填写下表。

V	j=0	1	2	3	4	5
i=0	0	0	0	0	0	0
1	0					
2	0					
3	0					
4	0					

答：

V(0, j) = 0（0个物品），V(i, 0) = 0（承重量0）

V(i, j) = V(i-1, j)　　第 i 个物品不能装入，　j < wi　（超重）

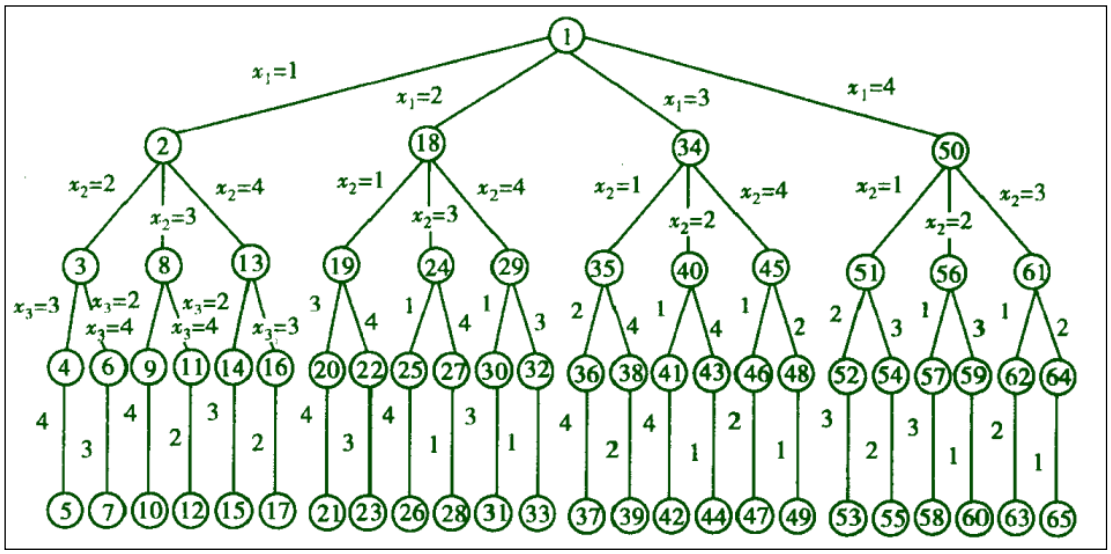
V(i, j) = max { v_i + V(i-1,j-w_j)　　，　V(i-1, j)　　}　　j > wi　（不超重）

i在最优子集中　　i不在最优子集中

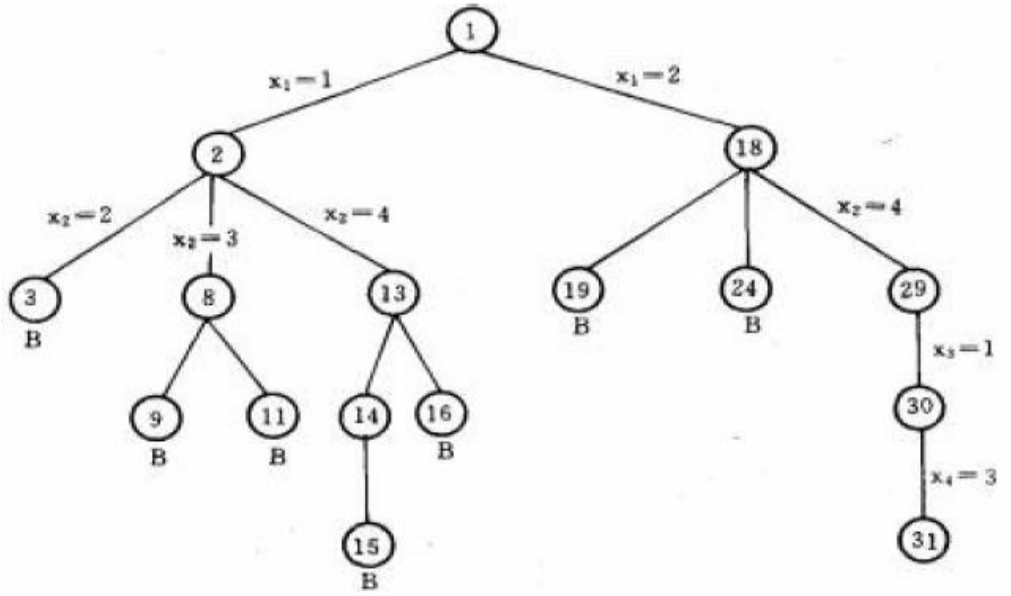
V	j=0	1	2	3	4	5
i=0	0	0	0	0	0	0
1	0	0	12	12	12	12
2	0	10	12	22	22	22
3	0	10	12	22	30	32
4	0	10	15	25	30	37

9. 请画出用回溯法解 **4 皇后问题的解空间树和搜索空间树：**

解空间树：



用回溯法的搜索空间树：



10.考虑用分支限界解 0-1 背包问题

给定 n 种物品和一背包。物品 i 的重量是 wi，其价值为 vi，背包的容量为 C。问应如何选择装入背包的物品，使得装入背包中物品的总价值最大?

示例：n=3, C=30, w={16, 15, 15}, v={45, 25, 25}

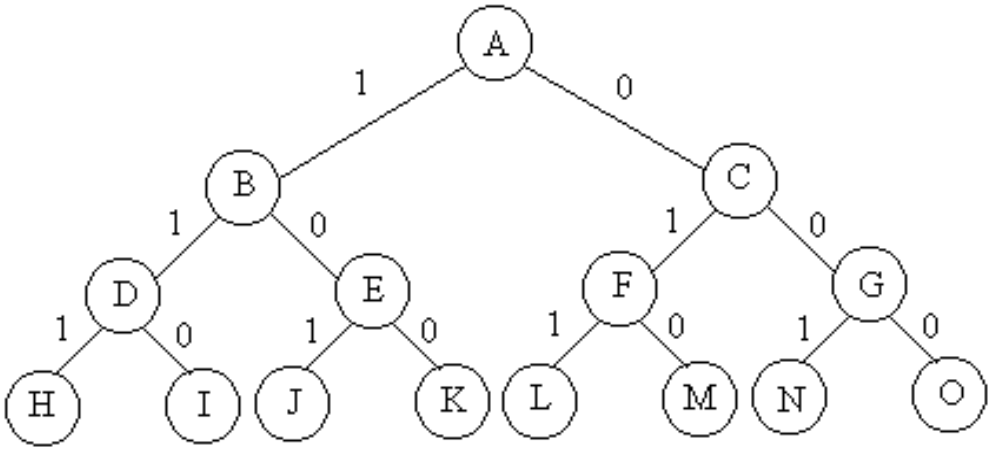
求：1、问题的解空间树

2、约束条件

2、如何剪枝？

解：

问题的解空间树：



约束条件： $$\sum_{i=1}^{n} w_i x_i \le c_1$$

如何剪枝？：

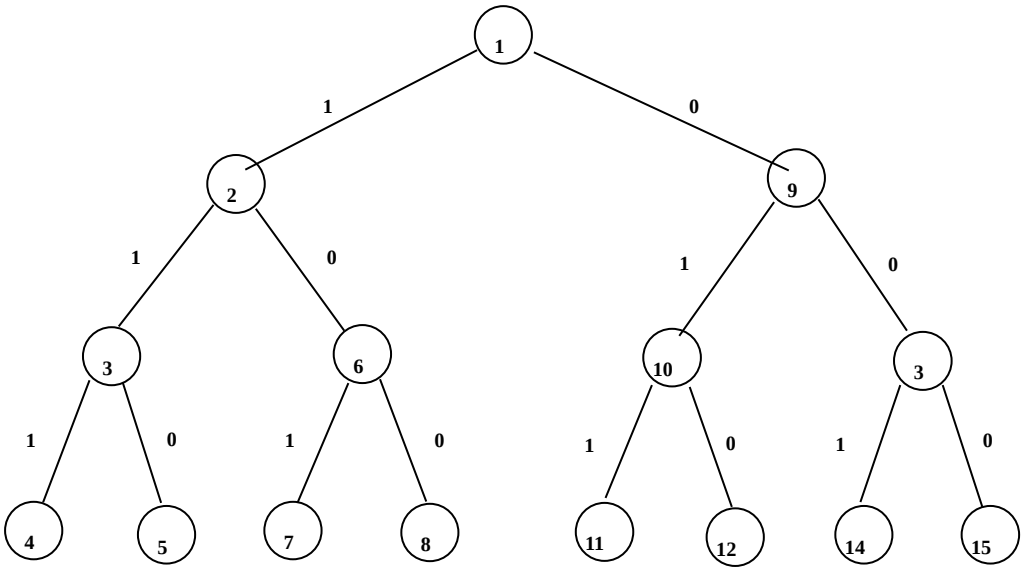
设 r 是当前尚未考虑的剩余物品价值总和；Cv 是当前价值；bestv 是当前最优价值。

当 r＋Cv≤bestv 时，可剪去右子树。

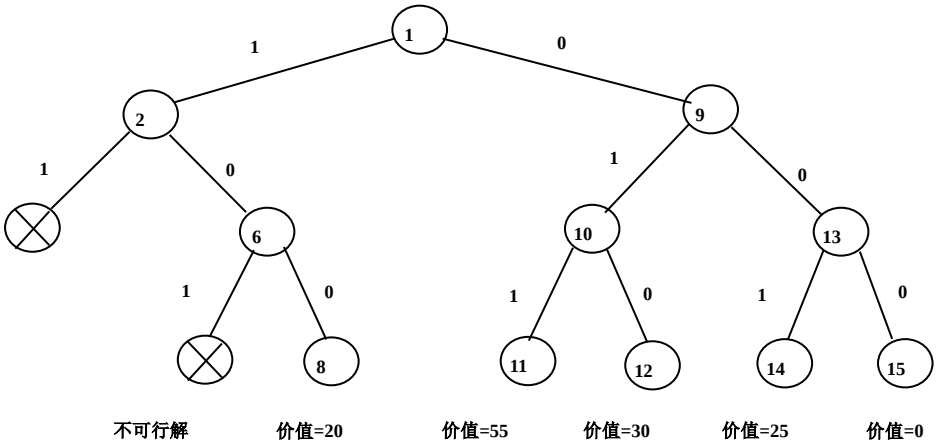
11，请画出用回溯法解n=3的0-1背包问题的解空间树和当三个物品的重量为{20, 15, 10}，价值为{20, 30, 25}，背包容量为25时搜索空间树。

答：

解空间树：


1
1
0
2
9
1
0
1
0
3
6
10
3
1
0
1
0
1
0
1
0
4
5
7
8
11
12
14
15

搜索空间树：


1
1
0
2
9
1
0
1
0
6
10
13
1
0
1
0
1
0
8
11
12
14
15
不可行解
价值=20
价值=55
价值=30
价值=25
价值=0