(5)

《算法分析与设计 》

院(系)_____ 班级_____ 学号_____ 姓名_____

题号	一	二	三	四	五	六	七	八	九	十	总分
得分											

装订线内不要答题

得分

一、选择题（每题2分，共20分）

1、算法可以有（ D ）个输出。

A 0个 B 0个或1个 C 0个或多个 D 1个或多个

2、以下关于算法和程序的描述中（ C ）是错误的。

A. 算法必须可终止，程序却没有这一限制。

B. 描述一个算法可以用自然语言、流程图、伪代码和程序设计语言。

C. 操作系统程序是使用计算机语言描述的一个算法。

D. 算法+数据结构=程序。

3、已知某算法所需时间的递推式为：$T(n)=2T(n/2)+n^2$，$T(1)=2$，则该算法的渐近时间复杂度为（ B ）。

A. $\Theta(n)$ B. $\Theta(n^2)$ C. $\Theta(\log n)$ D. $\Theta(n^2\log n)$

4、十字路口信号灯系统是一个程序，却不是算法，是因为算法必须具有（ B ）。

A 确定性 B 有穷性 C 可行性 D 输入

5、以下关于动态规划法的描述哪项是正确的（ C ）。

A. 动态规划法一般采用自顶向下方式进行计算，并且保存已求解子问题的最优解值。

B. 动态规划法一定具有多项式时间的渐近时间复杂度。

C. 动态规划法通常采用数学归纳法进行正确性证明。

D. 动态规划法比备忘录方法具有更好的时间界。

6、下面（ B ）问题不能使用贪心算法来解决。

A 单源最短路径问题 B N-皇后问题

C 最小代价生成树问题 D 一般背包问题

7、根据（ B ）可将分枝限界法分为FIFO、LIFO、LC分枝限界法三类。

A. 状态空间树中解空间的大小不同

B. 从活结点表中选择下一个扩展结点的次序不同

C. 求解可行解或最优解的目标不同

D. 构造的剪枝函数不同

8、设 $X=(x_1,x_2,\ldots,x_m)$ 和 $Y=(y_1,y_2,\ldots,y_n)$ 为两个序列，$Z=(z_1,z_2,\ldots,z_k)$ 是它们的最长公共子序列，则以下说法错误的是（ A ）。

A. 若 $x_m\neq y_n$ 且 $z_k\neq y_n$，则 Z 是 X_{m-1} 和 Y 的最长公共子序列；

B. 最长公共子序列具有最优子结构特性，且 Z 的取值不唯一。

C. 若 $x_m=y_n$，则 $z_k=x_m=y_n$，且 Z_{k-1} 是 X_{m-1} 和 Y_{n-1} 的最长公共子序列。

D. 求解 Z 可在多项式时间内完成计算，时间复杂度为 O(m*n)。

9、 以代价估计函数 $\hat{c}()$ 作为选择下一个扩展结点的评价函数，即总是选取 $\hat{c}()$ 值小的活结点作为下一个扩展结点，采用这种检索方法的分枝限界法属于（ C ）。

A.FIFO 分枝限界法　B. LIFO 分枝限界法　C. LC 分枝限界法　D. Heap 分枝限界法

10、下列关于对半搜索二叉判定树的说法中（ D ）是错误的。

A. 对半搜索所需的关键字值间的比较次数不会超过对半搜索二叉判定树的高度(不计失败结点)。

B. 对半搜索二叉判定树中，从根结点到每个内结点的一条路径代表成功搜索的一条比较路径。

C. 对半搜索二叉判定树中若搜索失败，则算法在外结点处终止。

D. 对半搜索二叉判定树一定是满二叉树。

得 分

二、填空题（每空 2 分，共 20 分）

1、使用剪枝函数的深度优先生成状态空间树中结点的求解方法称为回溯法，而分枝限界法则是利用 广度 优先生成结点。

2、不同的活结点表生成不同的分枝限界法，FIFO 分枝限界法的活结点表是先进先出队列，LIFO 分枝限界法的活结点表是堆栈，LC 分枝限界表的活结点表是 优先权队列 。

3、回溯法中一般用 蒙特卡罗 方法估计处理问题时状态空间树上实际生成的结点数。若随机选择一条路径 (x_0,x_1,x_2)，这条路径上不受限制的 x_0 取值有 5 个，不受限制的 x_1 取值有 4 个，不受限制的 x_2 取值有 2 个。则整个状态空间树上将实际生成的结点数估计为 m= 66 。

4、一个问题的最优解包含其子问题的最优解，则称此问题具有 最优子结构 性质，是该问题可以用动态规划算法或贪心算法求解的关键特征。

5 基于比较关键字值的两路合并排序最坏情况时间复杂度是 O(n log n) 。

6 图 G=(*V*, *E*)是一个无向带权连通图，若其边数相对较少，则能较高效率得到最小代价生成树的是 克鲁斯卡尔 算法。

7、分枝限界法生成状态空间树中的结点时，一旦一个活结点成为 扩展结点（E-结点） 后，算法将依次生成它的孩子结点并加入到活结点表中。

8、对半搜索的过程中形成一棵二叉判定树，则具有 n(n>0)个内结点的二叉判定树的高度(不计外结点)为 $\lfloor \log n \rfloor + 1$ 。

得 分

三、简答题（每题 10 分，共 30 分）

1、(10 分)确定下面画线语句的执行次数，并计算出其的渐近时间复杂度。

```
(1) for (int i=1;i<=n;i++)
      for (int j=1;j<=i;j++)
        for (int k=1;k<=j;k++)
          x++;
```

```
(2) i=1; x=0
    do{
        x++; i=2*i;
    } while i<n;
```

(1) 执行次数为 n(n+1)(n+2)/6　(3 分)

时间复杂度为 $O(n^3)$　(2 分)

(2) 执行次数为⌈log n⌉　(3 分)

时间复杂度为 O(log n)　(2 分)

2、(10 分) 请简要回答动态规划法与贪心法的异同点有哪些？

（1）共同点：

都是求解最优化问题；　(2 分)

都要求问题具有最优子结构性质。　(2 分)

（2）不同点：

1)、求解方式不同：　(3 分)

动态规划法：自底向上；

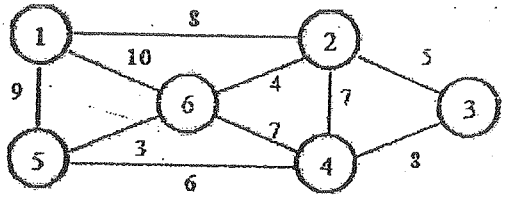
贪心法：自顶向下。以迭代的方式作出相继的贪心选择，每作一次贪心选择就将所求问题简化为一个规模更小的子问题。

2)、对子问题的依赖不同：　(3 分)

动态规划法：依赖于各子问题的解，所以只有在解出相关子问题后,才能作出选择；应使各子问题最优，才能保证整体最优；

贪心法：不依赖于子问题的解。仅在当前状态下作出最好选择，即局部最优选择，然后再去解作出这个选择后产生的相应的子问题。

3、(10 分) 给定如下图所示的带权无向连通图，请分别使用普里姆算法和克鲁斯卡尔算法画出最小代价生成树的构造过程。(要求普里姆算法以顶点 1 为起始点进行构造)



(1) 普里姆方法：

结点并入顺序：1,2,6,5,3,4（图略） (5 分)

(2) 克鲁斯卡尔方法：

边的并入顺序：(5,6)，(2,6)，(2,3)，(4,5)，(1,2)（图略） (5 分)

得 分

四、证明题（10 分）

假定 n 是 2 的幂，设 T(n) 由如下递推式定义，请利用迭代法扩展递推式以证明 $T(n) = O(n\log^2 n)$。

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

设 $n=2^k$，则

$T(n) = 2T(n/2) + n \log n$

$= 2 (2T(n/2^2) + (n/2)*\log(n/2)) + n \log n$

$= 2 (2 (2T(n/2^3) + (n/2^2)*\log(n/2^2)) + (n/2)*\log(n/2)) + n \log n$

.....

$= 2^k T(n/2^k) + n \log(n/2^{k-1}) + n \log(n/2^{k-2}) + \ldots\ldots + n \log(n/2^0)$

$= n + n\log 2 + n\log 2^2 + \ldots\ldots + n\log 2^k$

$= n + n*(1+2+\ldots\ldots+k)$

$= (n \log^2 n)/2 + (n \log n)/2 + n$

$= O(n\log^2 n)$

得 分

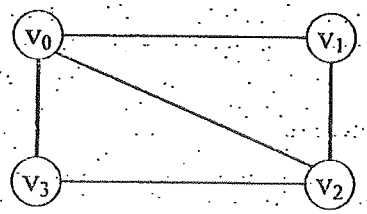
五、综合应用题（5 分+7 分+8 分，共 20 分）

设有背包问题实例 n=4，M=32，物品重量 $(w_0, w_1, w_2, w_3)=(6, 9, 10, 15)$，物品装入背包的收益为：$(p_0, p_1, p_2, p_3)=(8, 1, 2, 5)$。

（1） 假设每件物品可以分割，请给出利用贪心法求解这一实例的最优解，以及此时的最大收益。（请给出求解过程）

（2） 假设每件物品不可以分割，请给出利用动态规划法求解这一实例的最优解，以及此时的最大收益。（请给出求解过程）

（3） 假设给定编号分别为 a、b、c 的 3 个背包，现要求将上述 4 件物品装入这 3 个背包中，下图显示了 4 件物品的装包冲突图 G=（V，E），在图 G 中任意一个结点 $v_k \in V$ 表示重量为 w_k 的物品 $(k\in\{0, 1, 2, 3\})$，任意一条边 $(v_i, v_j) \in E$ 表示重量为 w_i 和 w_j 的两个物品不能放入同一个背包中，请利用回溯法求解在满足图 G 所示的约束条件下将上述 4 件物品装入这 3 个背包的所有可行的装包方案，给出问题的显示约束和隐式约束划分，并且画出对这一实例采用回溯法时实际生成的状态空间树。（提示：该问题的解可采用 4-元组 $X=(x_0, x_1, x_2, x_3)$ 表示，其中 x_i 表示重量为 w_i 的物品所装入背包的编号）



(1) $p_i / w_i = (4/3, 1/9, 1/5, 1/3)$，　　(2 分)

最优解：(1, 1/9, 1, 1)　　(2 分)

最大收益：P_{max} = 8+1*1/9+2+5=136 / 9　　(1 分)

(2) S^{-1} = {(0, 0)}, S_1^0 = {(6, 8)},
S^0 = {(0, 0), (6, 8)}, S_1^1 = {(9, 1), (15, 9)},
S^1 = {(0, 0), (6, 8), (15, 9)}, S_1^2 = {(10, 2), (16, 10), (25, 11)},
S^2 = {(0, 0), (6, 8), (15, 9), (16, 10), (25, 11)},
S_1^3 = {(15, 5), (21, 13), (30, 14), (31, 15), (40, 16)},

S^3 = {(0, 0), (6, 8), (15, 9), (16, 10), (21, 13), (30, 14), (31, 15)}　　(4 分)

因为(31, 15) ∈ S_1^3，所以 x_3 =1。计算(X-w_3, P-p_3)=(16, 10)，继续回溯，因为(16,10) ∈ S_1^2，

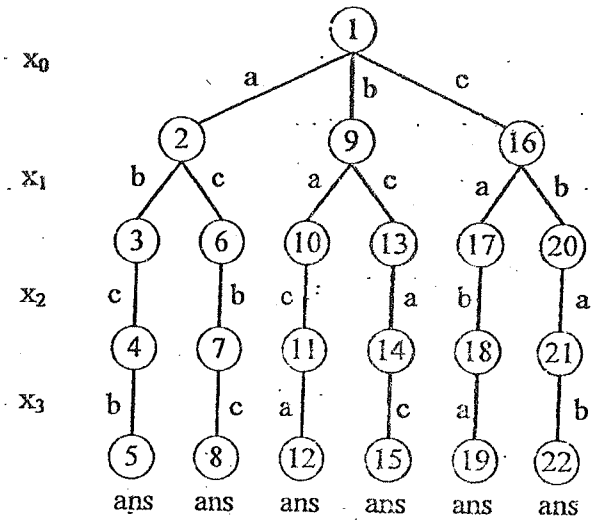
所以 x_2 =1。计算(X-w_2, P-p_2)=(6, 8)，继续回溯，因为(6, 8) ∈ S^0，所以 x_1 =0。再回溯，

因为(6, 8) ∈ S_1^0，所以 x_0 =1。因此，问题最优解为 (x_0, x_1, x_2, x_3) = (1, 0, 1, 1)　(2 分)

最大收益为 8+2+5=15。　(1 分)

(3) 显示约束：x_i ∈ {a, b, c}, 0≤i<4　　(1 分)

隐式约束：如果边（v_i, v_j）$\in E$，则 $x_i \neq x_j$　　（1 分）



因此，所有可行解为：

X = (a, b, c, b) = (a, c, b, c) = (b, a, c, a) = (b, c, a, c) = (c, a, b, a) = (c, b, a, b)　　（6 分）

《算法分析与设计》

院(系)_____ 班级_____ 学号_____ 姓名_____

题号	一	二	三	四	五	总 分
得分						

装订线内不要答题

得 分

一、选择题（每空 2 分，共 20 分）

1、计算机无法解决打印所有素数问题，其原因是解决该问题的算法违背了算法特性中的（ B ）。

A. 唯一性 B. 有穷性 C. 有 0 个或多个输入 D. 有输出

2、关键路径问题中，完成工程所需的最短时间是 AOE 网络中从开始结点到完成结点的（ B ）路径的长度。

A. 最短 B. 最长 C. 最优 D. 最差

3、下面算法的时间复杂度为（ D ）

```
i = 1; a = 0
while i <= n do {
    if w(i) > m return
    a = a + i
    i = i+1
}
```

A. $O(n^3)$ B. $O(n^2)$ C. O(logn) D. O(n)

4、在快速排序算法中引入随机过程的主要目的是（ C ）。

A. 改善确定性算法的平均时间

B. 保证算法总能在 O(nlogn)时间内结束

C. 避免了算法最坏情况下的发生

D. 改善了确定性算法最坏情形下的平均运行时间

5、有载重为 20 的背包，3 件物品的重量为：(w_0, w_1, w_2) =(15, 12, 18)，物品装入背包的收益为：(p_0, p_1, p_2)=(10, 9, 15)，物品可分割，最优装载方案的收益为（ D ）

A. 13.75 B. 15.67 C. 16.33 D. 16.5

6、两路合并排序的最坏情况时间复杂度为是（ C ）

A. $O(n^3)$ B. $O(n^2)$ C. O(nlogn) D. O(n)

7、有 5 名同学参加冬季越野赛跑，学号分别是 5，8，11，33，45，用分治法查找学号 33 的同学的过程中，依次访问道德学号是（ D ）

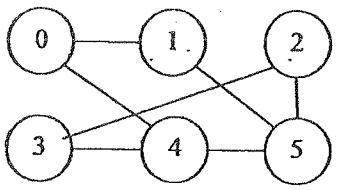
A. 5, 11, 33 B. 8, 33 C. 11, 45, 33 D. 11, 33

8、对 n 个元素构成的有序表进行对半搜索的二叉判定树高度为（ B ）

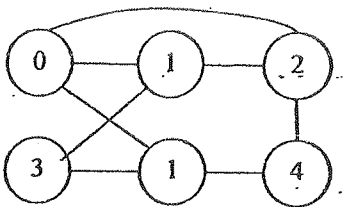
A. n　　B. $\lfloor \log n \rfloor + 1$　　C. $\lfloor n/2 \rfloor$　　D. 2^n-1

9、下面图中（ D ）不包含哈密顿环。

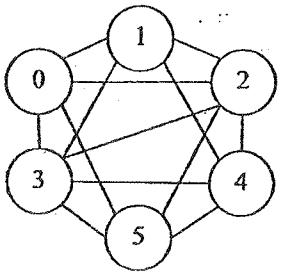
A.



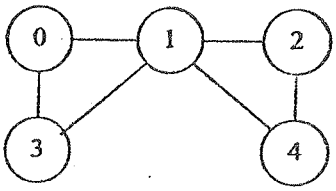
B.



C.



D.



10、下列（ A ）不是序列 X=<A,B,A,C,D,A,B>的子序列？

A. <B,C,B,A>　　B. <A,B,D,A>　　C. <A,A,C,B>　　D. <B,A,A,B>

得 分

二、连线题（每线 2 分，共 10 分）

根据各个算法的性质连线

子问题相互独立	子问题有重叠部分	子问题可以找到最优策略	广度优先	深度优先

回溯法	贪心算法	动态规划法	分支限界法	分治法

得 分

三、问答题（每题 18 分，共 54 分）

1、已知欧几里德算法（辗转相除法）可以求得两个数的最大公约数

1）试简要分析一下原理（提示，$a = k_1 c$, $b = k_2 c$, 且k_1，k_2互质）

因为k_1，k_2互质，所以两个数相减得到的余数是最大公约数的倍数，最后直到其中一个数为最大公约数。

2）试用迭代法写出辗转相除法的伪代码

```
void Swap(int &a, int &b)
{
        int c=a;a=b;b=c;
}
int Gcd(int m, int n)
{
      if (m==0) return n;
    if (n==0) return m;
    if (m>n) Swap(m, n);
   while (m>0)
    {
              int c=n%m;
         n=m;
          m=c;
    }
    return n;
}
```

2、已知 15 谜问题如图所示

1		3	4
5	2	7	8
9	6	10	12
13	14	11	15

1）试写出每个空格的 Less 函数

Less(1) = 0　Less(2) = 0　Less(3) = 1
Less(4) = 1　Less(5) = 1　Less(6) = 0
Less(7) = 1　Less(8) = 1　Less(9) = 1
Less(10) = 0　Less(11) = 0　Less(12) = 1
Less(13) = 1　Less(14) = 1　Less(15) = 0

2）试证明能达到目标状态的初始状态均满足判定要求$\sum_{k=1}^{16} Less(k) + i + j$为偶数（提示，考虑空格移动的方向分情况讨论奇偶性）

分别考虑空格左右移动和上下移动对判定函数奇偶性的影响，证明每次移动都不改变奇偶性，可得证。

3、给定字符串 X=(x_1,x_2,...,x_6)= abacbd 和 Y=(y_1,y_2,...,y_5)= baacd。

（1）请给出 LCS 算法求解最长公共子序列长度过程中数组 c 和数组 s 的值。

c[i][j]保存子序列 Xi=(x_1,...,x_i)和 Yj=(y_1,...,y_j)的最长公共子序列的长度。

s[i][j]记录 c[i][j]的值是由三个子问题 c[i-1][j-1]+1，c[i][j-1]和 c[i-1][j]中的哪一个计算得到的。若 c[i][j]=c[i-1][j-1]+1 时 s[i][j]=1； c[i][j]=c[i-1][j] 时 s[i][j]=2；c[i][j]=c[i][j-1]时 s[i][j]=3。

（2）求最长公共子序列的长度。

（3）请用箭头画出在 s 上执行 CLCS(int i, int j)算法构造最长公共子序列的追溯路径，并构造最长公共子序列。

附：CLCS(int i, int j)程序如下：

```
void CLCS(int i, int j)
{    if (i==0||j==0) return;
     if (s[i][j]==1)
     {
          CLCS(i-1,j-1);
          cout<<a[i];
     }
     else if (s[i][j]==2) CLCS(i-1,j);
          else CLCS(i,j-1);
}
```

解：（1）

		0	*b*	*a*	*a*	*c*	*d*
		0	1	2	3	4	5
0	0	0	0	0	0	0	0
a	1	0	0	1	1	1	1
b	2	0	1	1	1	1	1
a	3	0	1	2	2	2	2
c	4	0	1	2	2	3	3
b	5	0	1	2	2	3	3
d	6	0	1	2	2	3	4

c[i][j]数组

		0	*b*	*a*	*a*	*c*	*d*
		0	1	2	3	4	5
0	0	0	0	0	0	0	0
a	1	0	2	1	1	3	3
b	2	0	1	2	2	2	2
a	3	0	2	1	1	3	3
c	4	0	2	2	2	1	3
b	5	0	1	2	2	2	2
d	6	0	2	2	2	2	1

s[i][j]数组

（2）最长公共子序列长度为：4

（3）回溯路径如图所示。

最长公共子序列为：aacd

得 分

四、程序设计题（16分）

请用递归法输入斐波那契数列的前 20 个数（程序的缩进，用户友好性均是考察点）。

解：程序如下：

```
long Fib(int n)
{
    if (n<=1)
    {
        return n;
    }
    else
    {
        return     Fib(n-1)+Fib(n-2);
    }
}

int main(int argc, char* argv[])
{
    // 递归法
    for (int i=1; i<=20; i++)
    {
        printf("%d ", Fib(i));
    }
    return 0;
}
```

装订线内不要答题

《算法分析与设计 》

院(系)_____ 班级_____ 学号_____ 姓名_____

题号	一	二	三	四	五	总 分
得分						

自觉遵守考试规则，诚信考试，绝不作弊 装订线内不要答题

得 分

一、判断题（每题 2 分，共 10 分）

1、若求解的问题有较多重叠子问题，适合用分治法求解。 （ ）

2、最大集团判定问题是 NP 完全的。 （ ）

3、如果有两个定义在正数集上的正函数 f(n)和 g(n)满足 $f(n) = \Theta(g(n))$，则表示 g(n)既是 f(n)的一个上界，又是 f(n)的一个下界。 （T）

4、一般背包问题和 0-1 背包问题都可以用贪心法求解得到最优解。 （F）

5、分枝限界法中每个活结点可能多次成为当前扩展结点。 （F）

得 分

二、选择题（每题 2 分，共 20 分）

1、下列关于算法复杂度的说法，哪一个不正确（ B ）

A. 算法复杂度包括时间复杂度和空间复杂度两个方面。

B. 程序步数可以精确的反映程序运行的实际时间。

C. 同一个算法的时间复杂度又分为最好、最坏和平均情况三种。

D. 算法的时间复杂度不仅仅依赖于问题的规模，还与输入实例的初始状态有关。

2、下面划线程序步执行的时间复杂度为（ A ）。(n>1)

```
x=n;
y=0;
while (y<x) { y++; x--; }
```

A.O(n) B.$O(\sqrt{n})$ C.O(logn) D.$O(n^2)$

3、关于两路合并排序和快速排序算法，以下（ ）说法是错误的。

A. 若分解得到的子序列为空或只有一个元素时，认为其自然有序无需排序。

B. 两路合并排序的分解过程非常简单，每次只需将待排序的序列一分为二即可。

C. 快速排序中，一旦左、右两个子序列均已分别排序，则无需再进行额外处理，整个序列便自然成为了有序序列。

D. 两路合并排序的平均时间复杂度为 $O(n^2)$。

4、由若干权值分别为（1，3，3，4，7）的结点，按照贪心准则构造的两路最佳合并树的带权外路径长度为（ A ）

A. 40　　B. 43　　C. 45　　D. 48

5、斯特拉森矩阵乘法所需时间的递推式为：$T(n)=7T(n/2)+dn^2$，则该矩阵乘法的渐近时间复杂度为（　）

A. $\Theta(n^3)$　　B. $\Theta(n^{\log 7})$　　C. $\Theta(n^2)$　　D. $\Theta(\log n)$

6、n 个顶点的连通图至少有（ A ）条边。

A. n-1　　B. n　　C. n+1　　D. 0

7、备忘录方法是（　）的一个变种，以自顶向下直接递归的方式计算最优解，同时避免了相同子问题的重复求解。

A. 分治法　　B. 贪心法　　C. 动态规划法　　D. 回溯法

8、下列（ D ）是指数时间算法的渐近时间复杂度。

A. O(1)　　B. O(n)　　C. $O(n^3)$　　D. O(n!)

9、（ A ）问题是所有可在多项式时间内用确定算法求解的判定问题的集合。

A. P 类　　B. NP 类　　C. NP 难度　　D. NP 完全

10、第一个被证明是 NP 完全问题的是（　）。

A. CNF 可满足性问题　　B. 最大集团问题

C. 顶点覆盖问题　　D. 图着色数判定问题

得分

三、填空题（每空 2 分，共 20 分）

1、估算回溯法处理一个实例时，实际生成的结点数的方法叫 蒙特卡罗方法 。

2、贪心法是一种求解 最优化 问题的算法设计策略，每一步上用作决策依据的选择准则被称为 最优量度标准 。

3、分治法求解很自然导致一个 递归 算法。

4、若 $d_k[i][j]$表示从结点 i 到结点 j 的路径上，只允许包含结点编号不大于 k 的结点时，所有可能路径中的最短路径长度。则弗洛伊德算法中，求解带权有向图中任意一对结点间距离的递推式为 $d_k[i][j]=\min\{$ (3) , (4) $\}$

4、约束函数和限界函数统称为 剪枝函数 。

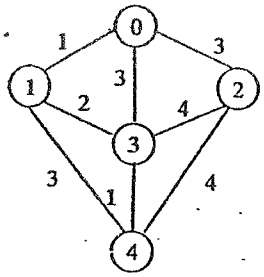
5、密码系统的安全性是基于 (6) 的，因此算法往往可以作为标准公布。密码体制可以从原理上分为 (7) 体制和 (8) 体制两类。

得分

四、简答题（每题 8 分，共 40 分）

1、对初始数据（3，6，1，2，8，8）执行快速排序，每次都选择左边第一个元素作为主元，请给出每一趟分划操作的结果，并用[ ]标识分划所得子序列的左、右边界。

2、带权的无向连通图 G 如下图所示：



（1）请画出从源点 0 开始，对图 G 执行 Prim 算法，得到最小代价生成树的步骤。

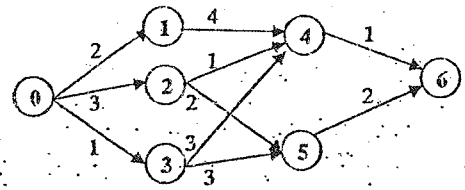
（2）请画出对图 G 执行 Kruskal 算法，得到最小代价生成树的步骤。

3、背包载重为 8，有 3 件物品的重量为(w_0,w_1,w_2)=（3,2,5），放入背包的收益分别为(p_0,p_1,p_2) = (6,6,5)。（注：最优解中的解分量 x_i 表示物品 i 放入背包的比例。）

（1）若物品可以分割，则最大收益是多少？最优解是什么？

（2）若物品不可以分割，则最大收益是多少？最优解是什么？

4、使用从前向后递推方法，求如图所示的多段图从源点 0 到汇点 6 的最短路径及路径长度。（请写出求解时的计算过程。）



5、NQueens 函数用回溯法求解 4-皇后问题的所有可行解，若随机选择的路径为(0,3,1)和（1,3,0,2），请用蒙特卡罗方法估计实际生成的状态空间树结点数。

得 分

五、算法程序题（每空 2 分，共 10 分）

请补充完整下面程序，完成折半插入排序。每次插入对象 V[i] 时，利用对半搜索在已排序好的对象序列中寻找 V[i] 的插入位置。

```
typedef int T;
void BInsSort ( T V[ ], int n ) {
  T temp;
  int Left, Right;
  for ( int i = 1; i < n; i++) {
        Left = 0;  Right = i-1;  temp = V[i];
        while ( Left <= Right ) {
            int mid = ____( i )____;
```

```
                if ( temp < V[mid] )    Right = ____(2)____;
            else        Left = ____(3)____;
        }
    for ( int k = i-1; k >= Left; k-- ) ____(4)____;
    ____(5)____ = temp;
  }
}
```

装订线内不要答题

自觉遵守考试规则，诚信考试，绝不作弊

《算法设计与分析》

院(系)_____ 班级_____ 学号_____ 姓名_____

题号	一	二	三	四	五	六	七	八	九	十	总分
得分											

得分

一、选择题（每空1分，共10分）

1、下列关于算法复杂度的说法，哪一个不正确（B）

A. 算法复杂度包括时间复杂度和空间复杂度两个方面。

B. 程序步数可以精确的反映程序运行的实际时间。

C. 同一个算法的时间复杂度又分为最好、最坏和平均情况三种。

D. 算法的时间复杂度不仅仅依赖于问题的规模，还与输入实例的初始状态有关。

2、快速排序在最坏情况下的时间复杂度是（D）

A. O(logn)　　B. O(n)　　C. O(nlogn)　　D. $O(n^2)$

3、问题能够用分治法求解的要素中不包含下列哪一个选项（B）

A. 问题能够按照某种方式分解成规模较小、相互独立且与原问题类型相同的子问题。

B. 问题具有重叠子问题性质。

C. 子问题足够小时可以直接求解。

D. 能将子问题的解组合成原问题的解。

4、对半搜索算法的成功搜索和失败搜索的时间复杂度分别为（A）

A. O(logn)　Θ(logn)　　B. O(logn)　O(logn)

C. Θ(logn)　O(logn)　　D. Θ(logn)　Θ(logn)

5、n个顶点的连通图至少有（A）条边。

A. n-1　　B. n　　C. n+1　　D. 0

6、设有5个有序文件分别包含（30,10,20,15,20）条记录，现通过两两合并将其合并成一个有序文件。（若每一次合并需读写两个文件的全部记录一次）则整个合并过程中至少需要读写总共（C）条记录。

A. 95　　B. 145　　C. 215　　D. 255

7、请将以下A-D选项

A. x[k]=i　　B. NQueens(0,n,x)　　C. Place(k,i,x)　　D. NQueens(k+1,n,x)

填入程序中合适的位置，使得程序能够用回溯法求解n-皇后问题的全部可行解：

(1)C　(2)A　(3)D　(4)B

```
bool Place(int k,int i,int *x)
{    for (int j=1;j<k;j++)
         if ((x[j]=i)||(abs(x[j]-i)==abs(j-k)))
             return false;
     return true;
}
void NQueens(int k,int n,int *x)
{   for (int i=0;i<n;i++)
    {   if(      (1)  C    )
        {     (2)  A    ;
              if (k==n-1)
              {   for (i=0;i<n;i++) cout<<x[i]<< " ";
                  cout<<endl;
              }
              else      (3)  D     ;
        }
    }
}
void NQueens(int n,int *x)
{      (4)  B     ;  }
```

得分

二、填空题（每空1分，共10分）

1、程序可以不满足算法的 有穷性 性质。

2、 划分操作 是快速排序的核心操作。

3、问题可用动态规划法或贪心法求解的关键特征是问题具有 最优子结构 性质。

4、用贪心法求解，找出 最优量度 标准是至关重要的。

5、对于问题的一个实例，<u>显式约束</u>规定了该问题的解空间，隐式约束给出了判定一个候选解是否为 可行解 的条件。

6、<u>　　</u>方法可以估计回溯法处理实例时状态空间树中实际生成的结点数。

7、根据从活结点中选择下一个扩展结点的不同次序，将分支限界法分为 FIFO分枝限界法、 LIFO 分枝限界法和 LC 分枝限界法三类。

得分	

三、判断题（每题1分，共10分）

1、简单的算法一定是高效的。 (X)

2、算法不能用流程图或伪代码来描述。 (X)

3、分治法求解很自然导致一个递归算法。 (√)

4、贪心算法每一步作出的选择是局部最优选择，而非整体最优选择。 (√)

5、贪心算法可以求解所有问题的整体最优解。 (X)

6、克鲁斯卡尔最小代价生成树算法，对边数较多（接近于完全图）的无向连通带权图有较高效率。 (X)

7、动态规划法在自底向上计算的过程中，保存子问题的解，因此可以避免计算重叠子问题。 (√)

8、回溯法运行的时间取决于状态空间树上所有问题状态的数目。 (√)

9、回溯法中每个活结点可能多次成为当前扩展结点。 (X)

10、分枝限界法采用深度优先的方式搜索问题的状态空间树。 (X)

得分	

四、证明题（10分）

设 f(n)、g(n)是定义在正数集上的正函数。证明：O(f(n))+O(g(n)) = O(max{f(n),g(n)})。

得分

五、简答题（每题10分，共20分）

1、比较动态规划法和贪心法的异同

2、比较分枝限界法与回溯法的异同

得 分

六、应用题（每题 10 分，共 40 分）

1、请写出对数据(6,8,2,4,6,3,5)进行快速排序的执行过程。（结果从小到大排列）

6 8 2 4 6 3 5

1. 6 5 2 4 3 6 8
2. 3 5 2 4 6 6 8
3. 4 2 5 3
4. 4 2 3 5
5. 2 4 3 5
6. 2 3 4 5

6 8 2 4 6 3 5

1. [6 5 2 4 6 3] 8
2. [3 5 2 4] 6 6 8
3. [4 2 5 3] 6 6 8
4. [4 2 3 5]
5. [2 4 3 5]
6. 2 [3 4]
7. 2 [3] 4 5
8. 2 3 4 8

2、设有一般背包问题实例 n=6，M=12，$(w_0,w_1,\ldots\ldots,w_5)=(2,5,7,4,1,3)$，物品装入背包的收益为 $(p_0,p_1,\ldots\ldots,p_5)=(8,12,7,6,3,6)$。求这一实例的最优解（6-元组形式）和最大收益。

$\left(\frac{p_0}{w_0}, \frac{p_1}{w_1}, \frac{p_2}{w_2}, \frac{p_3}{w_3}, \frac{p_4}{w_4}, \frac{p_5}{w_5}\right)=\left(4, \frac{12}{5}, 1, \frac{3}{2}, 3, 2\right)$

1 1 0 $\frac{1}{4}$ 1 1

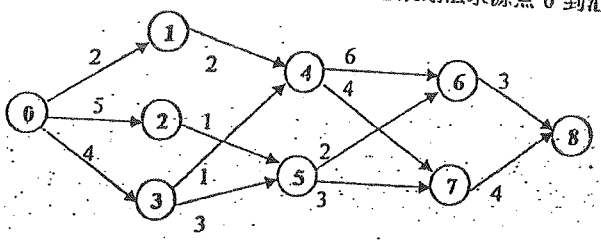
最优解

$(x_0,x_1,x_2,x_3,x_4,x_5)=(1,1,0,\frac{1}{4},1,1)$

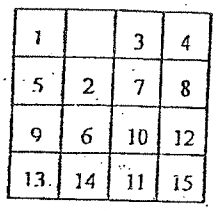
最大收益 $=8+12+\frac{3}{2}\times 4\times\frac{1}{4}+3+6=30.5$

15

3、对如图所示的带权有向无环图（AOE 网），用动态规划法求源点 0 到汇点 8 的关键路径。

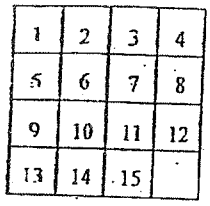


4、设一个 4×4 方形棋盘上给定 15 个号牌和一个空格的初始排列形式如图（a）：



1		3	4
5	2	7	8
9	6	10	12
13	14	11	15

（a）



1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

（b）

（1）从该状态出发，经过一系列合法的号牌移动，能否到达目标状态（b）？为什么？

（2）如果由初始状态（a）可以得到目标状态（b），请画出用 LC 分枝限界法检索目标结点时所实际生成的那部分状态空间树。（代价估计函数定义为 $\hat{c}(X)=f(X)+\hat{g}(X)$，其中 $f(X)$是从根到结点 X 的路径长度，$\hat{g}(X)$是不在其位的非空白牌数目。）

《算法设计与分析》

得分	

一、判断题（10分，共5题）

1、近似算法解最优化问题时需保证得到的结果在一定误差之内。（ √ ）

2、剪枝函数就是约束函数。（ × ）

3、蒙特卡罗方法可以精确得到状态空间树上实际生成的结点数。（ × ）

4、渐近时间复杂度使得可以在数量级上估计一个算法的执行时间。（ √ ）

5、一般称用于确定n个元素的排列满足某些性质的状态空间树为排列树。（ √ ）

得分	

二、填空题（20分，共10空）

1、辗转相除法又称 欧几里德算法 ，是一种求两个整数的 最大公约数 的算法。

2、很多情况下，可以通过考察一个程序中 关键操作 的执行次数，来估算算法的渐近时间复杂度。

3、如果 $f(n)=a^m n^m+a^{m-1}n^{m-1}+\cdots+a^1 n+a^0$ 是 m 次多项式，且 $a^m>0$，则 $f(n)=O(\ n^m\ )$。

4、贪心法在算法的每一步上根据 最优量度标准 选择分量，只需保证形成的部分解不违反约束条件，最终得到的n-元组必定是 最优解 。

5、Cook 定理表明 可满足性问题 是NP完全的。若问题 $Q\in NP$ 且Q是 NP难度 的，则称Q是NP完全的。

6、信息安全的目标是保护信息的机密性、 完整性 ，并具有 抗否认性 和可用性。

得分	

三、选择题（10分，共5题）

1、二分搜索算法的执行过程可以用一个（ B ）来描述。

A. 哈夫曼树　　B. 二叉判定树　　C. 堆栈　　D. 队列

2、实时系统最关心系统（ C ）情况下的时间复杂度。

A. 最好　　B. 平均　　C. 最坏　　D. 所有

3、下面关于备忘录方法的说法中（ A ）是错误的。

A. 备忘录方法自底向上求解。

B. 备忘录方法为每个已解过的子问题建立备忘录进行保存。

C. 备忘录方法是动态规划法的变形。

D. 当只有部分子问题需要求解时，选用备忘录方法较好。

4、若已知RSA算法中的公开密钥为{e,n}，私人密钥为{d,n}，M为明文分组，C是M对应的密文。则加密公式为（ A ）

A. $C=M^e \bmod n$　　B. $C=M^d \bmod n$　　C. $M=C^e \bmod n$　　D. $M=C^d \bmod n$

9

5、下面（ D ）不是针对 RSA 算法的攻击方式。

A. 大整数分解　　B. 时间攻击　　C. 中间人攻击　　D. 数字认证

得 分

四、问答题（50 分，共 5 题）

1、（1）请写出分治法的算法思想。

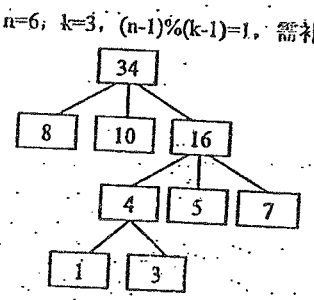
（2）分治法求解的问题通常得到一个递归算法，若算法的执行时间可以表示为 $T(n)=aT(n/b)+cn^k$，$T(1)=c$，请给出该算法渐近时间复杂度的解。

答：（1）分治法的算法思想是 1）将一个难以直接求解的复杂问题分解成若干个规模较小、相互独立，但类型相同的子问题。2）如果子问题还比较复杂而不能直接求解，还可以继续细分，直到子问题足够小，能够直接求解为止。3）最后，再通过组合子问题的解，来得到原始问题的完整解。

（2）$$T(n)=\begin{cases}\Theta(n^{\log_b a}) & \text{若} a>b^k \\ \Theta(n^k \log n) & \text{若} a=b^k \\ \Theta(n^k) & \text{若} a<b^k\end{cases}$$

2、请画出 W={1，3，5，7，8，10}的三路最佳合并树。

答：因为 n=6，k=3，(n-1)%(k-1)=1，需补齐一个虚结点。



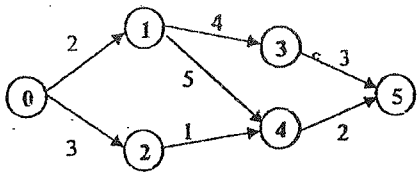
3、下图所示的 AOE 网中

（1）求每个事件（结点）i 可能的最早发生时间 earliest(i)和允许的最迟发生时间 latest(i)。

（2）求从源点 0 到汇点 5 的关键路径长度。

（3）找出图中所有的关键活动。

（4）写出图中所有的关键路径。



解：（1）

	0	1	2	3	4	5
earliest(i)	0	2	3	6	7	9
latest(i)	0	2	6	6	7	9

（2）关键路径长度：9

（3）关键活动：<0,1><1,3><1,4><3,5><4,5>

（4）关键路径：(0,1,3,5)和(0,1,4,5)

4、请用启发式方法计算 0/1 背包问题。背包载重为 M=6，有 4 件物品的重量为(w_0，w_1，w_2，w_3)=（3.3，2，1.2，5），放入背包的收益分别为（p_0，p_1，p_2，p_3）=（4，5，6，1）。背包最大收益是多少？最优解是什么？

（注：最优解中的解分量 x_i 表示物品 i 放入背包的比例。）

解：ProfLeft（0）=16，ProfLeft（1）=12，ProfLeft（2）=7，ProfLeft（3）=1，ProfLeft（4）=0。

物品 0，1 放入背包总重量 5.5<M=6，收益 9，得一可行解 L=9。

S^0={（0，0），（3.3，4）}　　x_0=0

S^1={（2，5），（5.3，9）}　　x_1=1

S^2={（3.2，11）}　　x_2=1

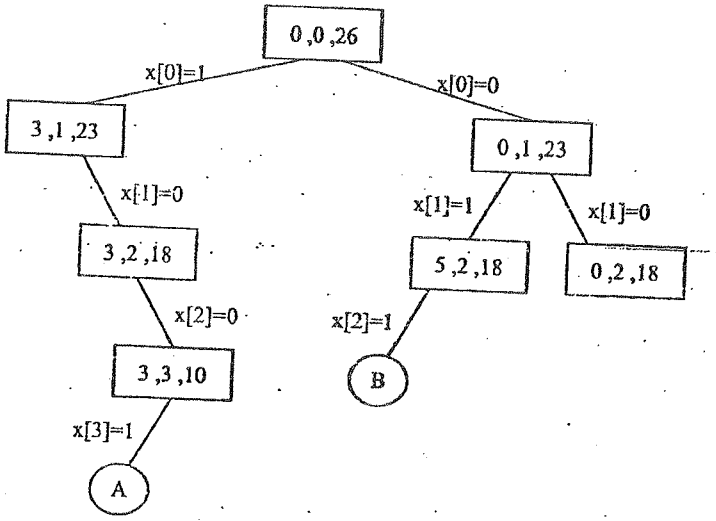
S^3={（3.2，11）}　　x_2=0

所以最大收益是 11，最优解是（0，1，1，0）。

5、设有 n=4 个正数的集合 W={w_0, w_1, w_2, w_3}=（3，5，8，10）和整数 M=13，求 W 中元素之和等于 M 的所有子集。画出对于这一实例由 SumOfSub 算法实际生成的那部分状态空间树，并给出所有可行解。

解：（1）实际生成的状态空间树：

29



（2）可行解有：（1，0，0，1）和（0，1，1，0）

得 分

五、算法设计题（10 分）

1、请利用快速排序算法的思想，用分治法设计一个算法选出 n 个元素中第 k 小的元素。请尽可能的说清楚你的算法，不用给出具体程序。

答：每一步根据一个随机选取的元素对输入数组进行递归划分，只对含有第 k 小元素的那部分子数组进行递归操作——即寻找 x[k]位置的所有者。

参考程序：

```
template <class Type>
Type RandomizedSelect(Type a[],int p,int r,int k)
    //在数组 a 中下标从 p 到 r 的范围内查找第 k 小的元素
{
    if (p==r) return a[p];     //a[p]即是第 k 小的元素
    int i=RandomizedPartition(a,p,r); //快速排序的一趟分划操作
    j=i-p+1; //前面子数组中（包括位置 i 处）元素总数
    if (k<=j)
        return RandomizedSelect(a,p,i,k);
    else
        return RandomizedSelect(a,i+1,r,k-j);
}
```

O(1) < O(log n) < O(n) < O(n log n) < O(n²) < O(n³)

时间复杂度　最好　最坏　平均

快速排序　O(n log n)　O(n²)　O(n log n)

合并排序：O(n log n)

两路合并：O(n log n)

对半搜索：

回记：

查找论文：

《算法分析与设计》

院(系)_____ 班级_____ 学号_____ 姓名_____

题号	一	二	三	四	五	六	七	八	九	十	总分
得分											

得分

一、选择题（每空1分，共10分）

1、下列关于算法的说法不正确的是（A）

A. 简单的算法一定是高效的。　B. 算法总能在执行有限步后终止。

C. 算法至少产生一个输出量。　D. 算法具有确定性。

2、两路合并排序算法最好、最坏、平均情况下的时间复杂度均为（　）

A. O(logn)　B. O(n)

C. O(nlogn)　D. $O(n^2)$

3、对半搜索算法的成功搜索和失败搜索的时间复杂度分别为（A）

A. O(logn)　Θ(logn)　B. O(logn)　O(logn)

C. Θ(logn)　O(logn)　D. Θ(logn)　Θ(logn)

4、有一般背包问题实例 n=6，M=12，$(w_0,w_1,\ldots,w_5)=(2,5,7,4,1,3)$，物品装入背包的收益为 $(p_0,p_1,\ldots,p_5)=(8,12,7,6,3,6)$，则这一实例的最大收益为（B）

A. 42　B. 30.5　C. 27.8　D. 25

5、以下对分枝限界法中活结点的叙述不正确的是（　）

A. 分枝限界法中的每个活结点只有一次机会成为扩展结点。

B. 活结点一旦成为扩展结点，依次生成该子结点后，自身就成为死结点。

C. 一个扩展结点生成的所有孩子结点，全部被加入活结点表中。

D. 从活结点表中选择下一个扩展结点时可依据的有 FIFO、LIFO 或 LC 原则。

6、十五谜问题适合手用（　）实现

A. 回溯法　B. FIFO 分枝限界法　C. LIFO 分枝限界法　D. LC 分枝限界法

7、请将以下 A-D 选项

A. x[k]=i　B. NQueens(0,n,x)　C. Place(k,i,x)　D. NQueens(k+1,n,x)

填入程序中合适的位置，使得程序能够用回溯法求解 n-皇后问题的全部可行解：

(1)_____ (2)_____ (3)_____ (4)_____

```
bool Place(int k,int i,int *x)
{   for (int j=1;j<k;j++)
```

```
        if ((x[j]=i)||(abs(x[j]-i)==abs(j-k)))
            return false;
    return true;
}
void NQueens(int k,int n,int *x)
{   for (int i=0;i<n;i++)
    {   if (      (1)      )
        {       (2)
            if (k==n-1)
            {   for (i=0;i<n;i++)  cout<<x[i]<<" ";
                cout<<endl;
            }
            else      (3)
        }
    }
}
void NQueens(int n,int *x)
{       (4)       ; }
```

得 分

二、填空题（每空1分，共10分）

1、算法的复杂度是指运行一个算法所需的 时间 和 空间 资源的量。

2、分治法求解很自然导致一个 递归 算法。

3、用贪心算法求解的问题一般都具有 最优子结构 性质和 贪心选择 性质。

4、设有5个有序文件分别包含（30,10,20,15,20）条记录，现通过两两合并将其合并成一个有序文件。（若每一次合并需读写两个文件的全部记录一次）则整个合并过程中至少需要读写总共 _____ 条记录。

5、 最优子结构 性质和 重叠子问题 性质是动态规划法求解的基本要素。

6、 状态空间树 是描述问题解空间的树形结构。

7、约束函数和限界函数统称为 剪枝函数。

得分	

三、判断题（每题1分，共10分）

1、操作系统既是一个程序，也是一个算法。 （X）

2、算法不能用自然语言来描述，而只能用程序设计语言来描述。 （X）

3、算法的时间复杂度不仅仅依赖于问题的规模，还与输入实例的初始状态有关。 （√）

4、程序步数可以确切的反映程序运行的实际时间。 （X）

5、如果有两个定义在正数集上的正函数 f(n)和 g(n)满足 $f(n) = \Theta(g(n))$，则表示 g(n)既是 f(n)的一个上界，又是 f(n)的一个下界。 （√）

6、一般背包问题和 0-1 背包问题都可以用贪心法求解得到最优解。 （X）

7、动态规划法自底向上逐步构造整体最优解的过程，不依赖于子问题的最优解。 （X）

8、回溯法运行的时间通常取决于状态空间树上实际生成的那部分问题状态的数目。 （√）

9、分枝限界法中每个活结点可能多次成为当前扩展结点。 （X）

10、如下图（a）所示，在 4×4 方形棋盘上给定了 15 个号牌和一个空格的初始排列。由该初始状态经过一系列合法的号牌移动，可以到达目标状态（b）。 （√）

1		3	4
5	2	7	8
9	6	10	12
13	14	11	15

（a）

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

（b）

33

得 分

四、证明题（10 分）

证明：n^2 是 3logn+2*n+n^2 的上界。

得 分

五、简答题（每题 10 分，共 20 分）

1、请比较分治策略中的合并排序和快速排序算法的不同。

V

2、比较动态规划法和分治法的异同

得 分

六、应用题（每题 10 分，共 40 分）

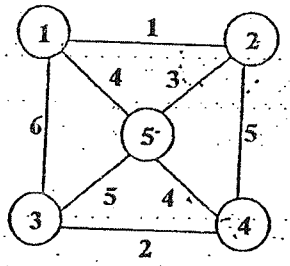
1、请写出对数据(6,5,4,3,2,1)进行快速排序的执行过程。(结果从小到大排列)

6 5 4 3 2 1

1. 1 5 4 3 2 6 ∞
2. 1 2 4 3 5 6 ∞
3. 1 2 3 4 5 6 ∞

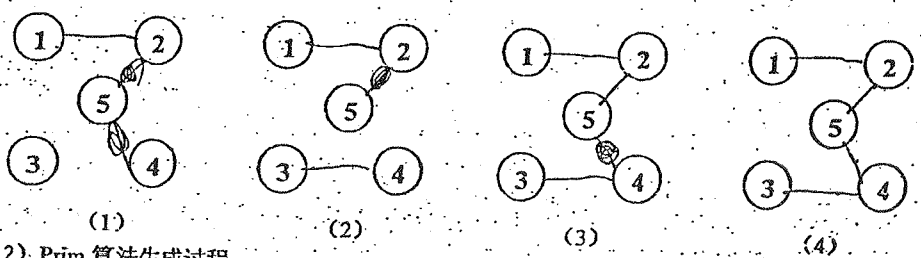
1. [1 5 4 3 2] 6
2. 1 [5 4 3 2] 6
3. 1 2 [4 3] [5 6]
4. 1 2 [3] 4 5 6
5. 1 2 3 4 5 6

2、对右图所示的一个无向连通带权图

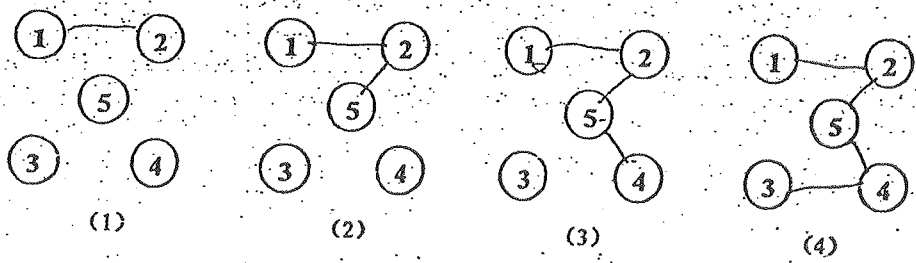


(1) 请在下面图中分别画出用 Kruskal 算法和 Prim 算法构造它的一棵最小代价生成树的过程。(从结点 1 出发)

1) Kruskal 算法生成过程：



2) Prim 算法生成过程：

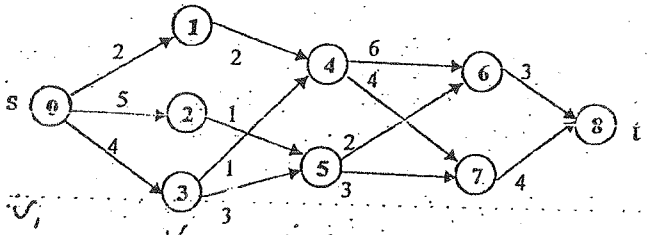


35

（2）对于边数相对较多（即比较接近于完全图）的无向连通带权图，比较适合于用哪种方法求解？对边数较少的无向连通带权图有较高效率的又是哪一种算法？

P　　K

3、对如图所示的多段图，采用向后递推和向前递推两种动态规划算法，求多段图中从源点 s 到汇点 t 的最短路径及路径长度。



V_1　V_2　V_3　V_4　V_5

向后：cost(5,8)=6

cost(4,6)=3　cost(4,7)=4

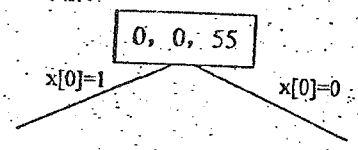
cost(3,4)=min{6+3, 4+4}=8　cost(3,5)=min{2+3, 3+4}=5

cost(2,1)=2+8=10　cost(2,5)=6　cost(3,5)=5

cost(1,0)=min{2+10, 5+6, 4+8}=11

向后：

4、设有 n=4 个正数的集合 $W=\{w_0,w_1,w_2,w_3\}=\{7,11,13,24\}$ 和整数 M=31，求 W 的所有满足条件的子集，使子集中的正数之和等于 M。请画出用回溯法求解的状态空间树。（采用固定长度 4-元组表示解）



37

一、选择题

1、 A
2、 C
3、 A
4、 B
5、 C
6、 D
7、 (1) C
(2) A
(3) D
(4) B

二、填空题

1、时间　　空间（顺序可颠倒）
2、递归
3、贪心选择　　最优子结构（顺序可颠倒）
4、215
5、最优子结构　　子问题重叠（顺序可颠倒）
6、状态空间树
7、剪枝函数

三、判断题

× × √ × √ × × √ × √

四、证明题

证明：即要证明 $31\log n+2*n+n^2=O(n^2)$。

取 $c=3, n_0=2$。　　（c 和 n_0 的取值可行各占 3 分，共 6 分）

当 $n\geq 2$ 的时候有 $31\log n+2*n+n^2\leq 3n^2$　　（2 分）

则 n^2 是 $31\log n+2*n+n^2$ 的上界。即：$31\log n+2*n+n^2=O(n^2)$。　　（写出其中任一句，即得 2 分）

五、简答题

1、答：

问题分解过程：　　（5 分）

- 合并排序——将序列一分为二即可。（十分简单）
- 快速排序——需调用 Partition 函数将一个序列划分为子序列。（分解方法相对较困难）

子问题解合并得到原问题解的过程：　　（5 分）

- 合并排序——需要调用 Merge 函数来实现。（Merge 函数时间复杂度为 O(n)）
- 快速排序——一旦左、右两个子序列都已分别排序，整个序列便自然成为有序序列。（异常简单，几乎无须额外的工作，省去了从子问题解合并得到原问题解的过程）

2、答：

共同点：　　（2 分）

将待求解的问题分解成若干子问题，先求解子问题，然后再从这些子问题的解得到原问题的解。

不同点：

1、适合于用动态规划法求解的问题，分解得到的各子问题往往不是相互独立的；而分治法中子问题相互独立。　　（4 分）

2、动态规划法用表保存已求解过的子问题的解，再次碰到同样的子问题时不必重新求

解，而只需查询答案，故可获得多项式级时间复杂度，效率较高；而分治法中对于每次出现的子问题均求解，导致同样的子问题被反复求解，故产生指数增长的时间复杂度，效率较低。

（4 分）

六、应用题

1、

答：

6 5 4 3 2 1 ∞

[1 5 4 3 2] 6 ∞

1 [5 4 3 2] 6 ∞

1 [2 4 3] 5 6 ∞

1 2 [4 3] 5 6 ∞

1 2 [3] 4 5 6 ∞

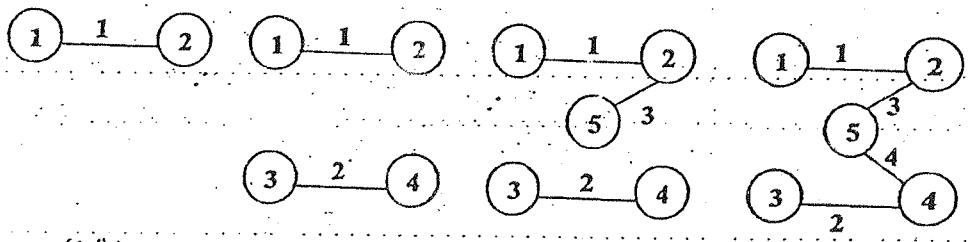
1 2 3 4 5 6 ∞

（求解形式正确占 1 分，哨兵∞占 1 分，最后排序结果正确占 1 分）

（排序过程占 7 分,错一步扣 1 分）

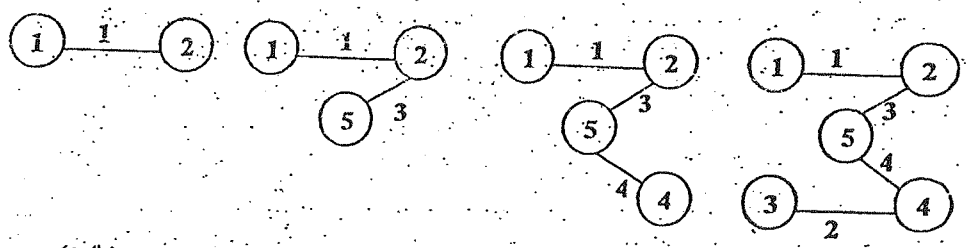
2、答：(1)

1）Kruskal 算法：



（1 分） （1 分） （1 分） （1 分）

2）Prim 算法：



（1 分） （1 分） （1 分） （1 分）

(2) 边数多的适用 Prim 算法，边数少的适用 Kruskal 算法。

（1 分） （1 分）

3、答：从后向前递推：

cost(8)=0; cost(7)=4; cost(6)=3; d(7)=d(6)=8;

cost(5)=min{2+cost(6),3+cost(7)}=5; d(5)=6;

cost(4)=min{6+cost(6),4+cost(7)}=8; d(4)=7;

cost(3)=min{1+cost(4),3+cost(5)}=8; d(3)=5;

cost(2)=1+cost(5)=6; d(2)=5;

cost(1)=2+cost(4)=10; d(1)=4;

cost(0)=min{2+cost(1),5+cost(2),4+cost(3)}=11; d(0)=2;

因此最短路径长度为 11；最短路径为（0,d(0)=2,d(2)=5,d(5)=6,d(6)=8）。

从前向后递推：

Bcost(0)=0; Bcost(1)=2; Bcost(2)=5; Bcost(3)=4; d(1)=d(2)=d(3)=0;

Bcost(4)=min{2+Bcost(1),1+Bcost(3)}=4; d(4)=1;

Bcost(5)=min{1+Bcost(2),3+Bcost(3)}=6; d(5)=2;

Bcost(6)=min{6+Bcost(4),2+Bcost(5)}=8; d(6)=5;

Bcost(7)=min{4+Bcost(4),3+Bcost(5)}=8; d(7)=4;

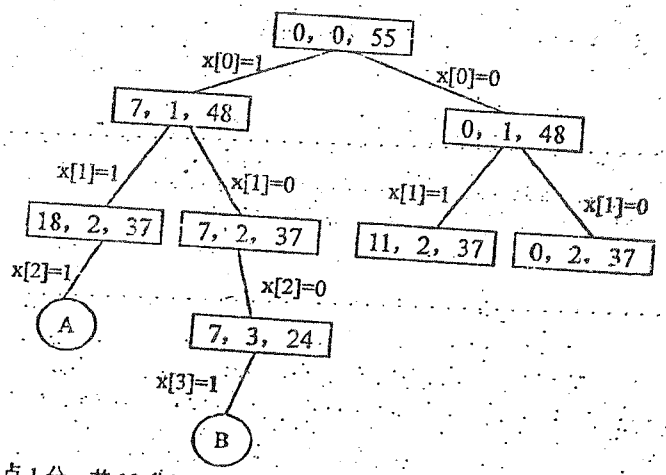
Bcost(8)=min{3+Bcost(6),4+Bcost(7)}=11; d(8)=6;

因此最短路径长度为 11；最短路径为（d(2)=0,d(5)=2,d(6)=5,d(8)=6,8）。

（向前、向后递推过程各占 3 分，共 6 分。其中向前、向后递推形式正确占 1 分，求值正确占 1 分，求值时递推表达式正确占 1 分）

（最短路径长度值占 2 分；向前、向后求得最短路径各占 1 分，共 2 分）

4、答：（w0,w1,w2,w3）=（7,11,13,24），M=31



（每个结点 1 分，共 10 分）

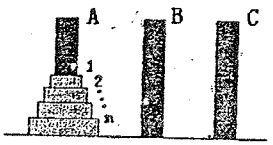
《算法分析与设计》期末复习题

一、 选择题

1.应用 Johnson 法则的流水作业调度采用的算法是（D）

A. 贪心算法　　B. 分支限界法　　C.分治法　　D. 动态规划算法

2.Hanoi 塔问题如下图所示。现要求将塔座 A 上的的所有圆盘移到塔座 B 上，并仍按同样顺序叠置。移动圆盘时遵守 Hanoi 塔问题的移动规则。由此设计出解 Hanoi 塔问题的递归算法正确的为：（B）



Hanoi 塔

```
A. void hanoi(int n, int A, int C, int B)
   {
      if (n > 0)
      {
         hanoi(n-1, A, C, B);
         move(n, a, b);
         hanoi(n-1, C, B, A);
      }
```

```
B. void hanoi(int n, int A, int B, int C)
   {
      if (n > 0)
      {
         hanoi(n-1, A, C, B);
         move(n, a, b);
         hanoi(n-1, C, B, A);
      }
```

```
C. void hanoi(int n, int C, int B, int A)
   {
      if (n > 0)
      {
         hanoi(n-1, A, C, B);
         move(n, a, b);
         hanoi(n-1, C, B, A);
      }
```

```
D. void hanoi(int n, int C, int A, int B)
   {
      if (n > 0)
      {
         hanoi(n-1, A, C, B);
         move(n, a, b);
         hanoi(n-1, C, B, A);
      }
```

3. 动态规划算法的基本要素为（C）

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

4. 算法分析中，记号O表示（B），记号Ω表示（A），记号Θ表示（D）。

A.渐进下界

B.渐进上界

C.非紧上界

D.紧渐进界

E.非紧下界

5. 以下关于渐进记号的性质是正确的有：（A）

A. $f(n)=\Theta(g(n)), g(n)=\Theta(h(n)) \Rightarrow f(n)=\Theta(h(n))$

B. $f(n)=O(g(n)), g(n)=O(h(n)) \Rightarrow h(n)=O(f(n))$

C. $O(f(n))+O(g(n))=O(\min\{f(n), g(n)\})$

D. $f(n)=O(g(n)) \Leftrightarrow g(n)=O(f(n))$

6. 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）

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

7. 回溯法在问题的解空间树中，按（D）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

8. 分支限界法在问题的解空间树中，按（A）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

9. 程序块（A）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);
      }
}
```

10. 回溯法的效率不依赖于以下哪一个因素？（C ）

A.产生 x[k]的时间；

B.满足显约束的 x[k]值的个数；

C.问题的解空间的形式；

D.计算上界函数 bound 的时间；

E.满足约束函数和上界函数约束的所有 x[k]的个数。

F.计算约束函数 constraint 的时间；

11. 常见的两种分支限界法为（D）

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；

12. k 带图灵机的空间复杂性 S(n)是指（B）

A. k 带图灵机处理所有长度为 n 的输入时，在某条带上所使用过的最大方格数。

B. k 带图灵机处理所有长度为 n 的输入时，在 k 条带上所使用过的方格数的总和。

C. k 带图灵机处理所有长度为 n 的输入时，在 k 条带上所使用过的平均方格数。

D. k 带图灵机处理所有长度为 n 的输入时，在某条带上所使用过的最小方格数。

13. NP 类语言在图灵机下的定义为（D）

A. NP={L|L 是一个能在非多项式时间内被一台 NDTM 所接受的语言}；

B. NP={L|L 是一个能在多项式时间内被一台 NDTM 所接受的语言}；

C. NP={L|L 是一个能在多项式时间内被一台 DTM 所接受的语言}；

D. NP={L|L 是一个能在多项式时间内被一台 NDTM 所接受的语言}；

14. 记号 O 的定义正确的是（A）。

A. $O(g(n)) = \{ f(n) \mid$ 存在正常数 c 和 n0 使得对所有 $n \geq n_0$ 有：$0 \leq f(n) \leq cg(n) \}$；

B. $O(g(n)) = \{ f(n) \mid$ 存在正常数 c 和 n0 使得对所有 $n \geq n_0$ 有：$0 \leq cg(n) \leq f(n) \}$；

C. $O(g(n)) = \{ f(n) \mid$ 对于任何正常数 c>0，存在正数和 $n_0>0$ 使得对所有 $n \geq n_0$ 有：$0 \leq f(n) < cg(n) \}$；

D. $O(g(n)) = \{ f(n) \mid$ 对于任何正常数 c>0，存在正数和 $n_0>0$ 使得对所有 $n \geq n_0$ 有：$0 \leq cg(n) < f(n) \}$；

15. 记号 Ω 的定义正确的是（B）。

A. $O(g(n)) = \{ f(n) \mid$ 存在正常数 c 和 n0 使得对所有 $n \geq n_0$ 有：$0 \leq f(n) \leq cg(n) \}$；

B. $O(g(n)) = \{ f(n) \mid$ 存在正常数 c 和 n0 使得对所有 $n \geq n_0$ 有：$0 \leq cg(n) \leq f(n) \}$；

C. $(g(n)) = \{ f(n) \mid$ 对于任何正常数 c>0，存在正数和 $n_0>0$ 使得对所有 $n \geq n_0$ 有：$0 \leq f(n) < cg(n) \}$；

D. $(g(n)) = \{ f(n) \mid$ 对于任何正常数 c>0，存在正数和 $n_0>0$ 使得对所有 $n \geq n_0$ 有：$0 \leq cg(n) < f(n) \}$；

二、 填空题

1. 下面程序段的所需要的计算时间为（ $O(n^2)$ ）。

```
int MaxSum(int n, int *a, int &besti, int &bestj)
{
int sum=0;
for(int i=1;i<=n;i++){
    int thissum=0;
    for(int j=i;j<=n;j++){
        thissum+=a[j];
        if(thissum>sum)  {
            sum=thissum;
            besti=i;
            bestj=j;
        }
    }
}
    return sum;
}
```

2. 有 11 个待安排的活动，它们具有下表所示的开始时间与结束时间，如果以贪心算法求解这些活动的最优安排（即为活动安排问题：在所给的活动集合中选出最大的相容活动子集合），得到的最大相容活动子集合为活动（ {1，4，8，11} ）。

i	1	2	3	4	5	6	7	8	9	10	11
S[i]	1	3	0	5	3	5	6	8	8	2	12
f[i]	4	5	6	7	8	9	10	11	12	13	14

3. 所谓贪心选择性质是指（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）。
4. 所谓最优子结构性质是指（问题的最优解包含了其子问题的最优解）。
5. 回溯法是指（具有限界函数的深度优先生成法）。
6. 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（O(h(n)）)。

7. 回溯法的算法框架按照问题的解空间一般分为（子集树）算法框架与（排列树）算法框架。

8. 用回溯法解 0/1 背包问题时，该问题的解空间结构为（子集树）结构。

9. 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（排列树）结构。

10. 用回溯法解 0/1 背包问题时，计算结点的上界的函数如下所示，请在空格中填入合适的内容：

```
Typep Knap<Typew, Typep>::Bound(int i)
{// 计算上界
   Typew cleft = c - cw;  // 剩余容量
   Typep b = cp;          // 结点的上界
   // 以物品单位重量价值递减序装入物品
   while (i <= n && w[i] <= cleft) {
      cleft -= w[i];
      b += p[i];
      i++;
      }
   // 装满背包
   if (i <= n)  (b += p[i]/w[i] * cleft);
   return b;
}
```

11. 用回溯法解布线问题时，求最优解的主要程序段如下。如果布线区域划分为 n×m 的方格阵列，扩展每个结点需 O(1)的时间，L 为最短布线路径的长度，则算法共耗时（ O(mn) ），构造相应的最短距离需要（O(L)）时间。

```
for (int i = 0; i < NumOfNbrs; i++) {
      nbr.row = here.row + offset[i].row;
      nbr.col = here.col + offset[i].col;
      if (grid[nbr.row][nbr.col] == 0) {
         // 该方格未标记
         grid[nbr.row][nbr.col]
            = grid[here.row][here.col] + 1;
         if ((nbr.row == finish.row) &&
            (nbr.col == finish.col)) break; // 完成布线
      Q.Add(nbr);}
      }
```

12. 用回溯法解图的 m 着色问题时，使用下面的函数 OK 检查当前扩展结点的每一个儿子所相应的颜色的可用性，则需耗时（渐进时间上限）（O（mn））。

```
Bool Color::OK(int k)
{//
  for(int j=1;j<=n;j++)
    if((a[k][j]= =1)&&(x[j]= =x[k])) return false;
  return true;
}
```

13. 旅行售货员问题的解空间树是（排列树）。

三、 证明题

1. 一个分治法将规模为 n 的问题分成 k 个规模为 n / m 的子问题去解。设分解阀值 n0=1，且 adhoc 解规模为 1 的问题耗费 1 个单位时间。再设将原问题分解为 k 个子问题以及用 merge 将 k 个子问题的解合并为原问题的解需用 f(n) 个单位时间。用 T(n) 表示该分治法解规模为 |P|=n 的问题所需的计算时间，

则有：$T(n)=\begin{cases} O(1) & n=1 \\ kT(n/m)+f(n) & n>1 \end{cases}$

通过迭代法求得 T（n）的显式表达式为：$T(n)=n^{\log_m k}+\sum_{j=0}^{\log_m n-1} k^j f(n/m^j)$

试证明 T（n）的显式表达式的正确性。

2. 举反例证明 0/1 背包问题若使用的算法是按照 p_i/w_i 的非递减次序考虑选择的物品，即只要正在被考虑的物品装得进就装入背包，则此方法不一定能得到最优解（此题说明 0/1 背包问题与背包问题的不同）。

证明：举例如：p={7,4,4},w={3,2,2},c=4 时，由于 7/3 最大，若按题目要求的方法，只能取第一个，收益是 7。而此实例的最大的收益应该是 8，取第 2，3 个。

3. 求证：O(f(n))+O(g(n)) = O(max{f(n),g(n)}) 。

证明：对于任意 f1(n)∈ O(f(n)) ，存在正常数 c1 和自然数 n1，使得对所有 n≥n1，有 f1(n)≤ c1f(n) 。

类似地，对于任意 g1(n) ∈ O(g(n))，存在正常数 c2 和自然数 n2，使得对所有 n≥n2，有 g1(n) ≤c2g(n) 。

令 c3=max{c1, c2}， n3 =max{n1, n2}，h(n)= max{f(n),g(n)} 。

则对所有的 n ≥ n3，有

$$\begin{aligned} f1(n)+g1(n) &\le c1f(n)+c2g(n) \\ &\le c3f(n)+c3g(n) \\ &= c3(f(n)+g(n)) \\ &\le c32\max\{f(n),g(n)\} \\ &= 2c3h(n) = O(\max\{f(n),g(n)\}). \end{aligned}$$

4. 求证最优装载问题具有贪心选择性质。

（最优装载问题：有一批集装箱要装上一艘载重量为 c 的轮船。其中集装箱 i 的重量为 Wi。最优装载问题要求确定在装载体积不受限制的情况下，将尽可能多的集装箱装上轮船。 设集装箱已依其重量从小到大排序，$(x_1, x_2, \cdots, x_n)$ 是最优装载问题的一个最优解。又设 $k=\min_{1\le i\le n}\{i \mid x_i=1\}$ 。如果给定的最优装载问题有解，则有 $1\le k\le n$ 。）

证明：

四、 解答题

1. 机器调度问题。

问题描述：现在有 n 件任务和无限多台的机器，任务可以在机器上得到处理。每件任务的开始时间为 s_i，完成时间为 f_i，$s_i<f_i$ 。$[s_i, f_i]$为处理任务 i 的时间范围。两个任务 i，j 重叠指两个任务的时间范围区间有重叠，而并非指 i，j 的起点或终点重合。例如：区间[1，4]与区间[2，4]重叠，而与[4，7]不重叠。一个可行的任务分配是指在分配中没有两件重叠的任务分配给同一台机器。因此，在可行的分配中每台机器在任何时刻最多只处理一个任务。最优分配是指使用的机器最少的可行分配方案。

问题实例：若任务占用的时间范围是{[1，4]，[2，5]，[4，5]，[2，6]，

[4，7]}，则按时完成所有任务最少需要几台机器？（提示：使用贪心算法）画出工作在对应的机器上的分配情况。

2. 已知非齐次递归方程：$\begin{cases} f(n)=bf(n-1)+g(n) \\ f(0)=c \end{cases}$，其中，b、c 是常数，g(n)是 n 的某一个函数。则 f(n)的非递归表达式为：$f(n)=cb^n+\sum_{i=1}^{n} b^{n-i}g(i)$。

现有 Hanoi 塔问题的递归方程为：$\begin{cases} h(n)=2h(n-1)+1 \\ h(1)=1 \end{cases}$，求 h(n)的非递归表达式。

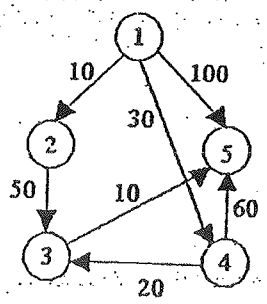
解：利用给出的关系式，此时有：b=2, c=1, g(n)=1，从 n 递推到 1，有：

$$\begin{aligned} h(n) &= cb^{n-1}+\sum_{i=1}^{n-1} b^{n-1-i}g(i) \\ &= 2^{n-1}+2^{n-2}+\ldots+2^2+2+1 \\ &= 2^n-1 \end{aligned}$$

3. 单源最短路径的求解。

问题的描述：给定带权有向图（如下图所示）G =(V, E)，其中每条边的权是非负实数。另外，还给定 V 中的一个顶点，称为源。现在要计算从源到所有其它各顶点的最短路长度。这里路的长度是指路上各边权之和。这个问题通常称为单源最短路径问题。

解法：现采用 Dijkstra 算法计算从源顶点 1 到其它顶点间最短路径。请将此过程填入下表中。



迭代	S	u	dist[2]	dist[3]	dist[4]	dist[5]
初始	{1}	-	10	maxint	30	100
1						
2						
3						
4						

4. 请写出用回溯法解装载问题的函数。

装载问题：有一批共 n 个集装箱要装上 2 艘载重量分别为 c1 和 c2 的轮船，其中集装箱 i 的重量为 wi，且 $\sum_{i=1}^{n} w_i \le c_1 + c_2$。装载问题要求确定是否有一个合理的装载方案可将这 n 个集装箱装上这 2 艘轮船。如果有，找出一种装载方案。

```
解: void backtrack (int i)
    {// 搜索第 i 层结点
      if (i > n)  // 到达叶结点
      更新最优解 bestx, bestw; return;
      r -= w[i];
      if (cw + w[i] <= c) {// 搜索左子树
        x[i] = 1;
        cw += w[i];
        backtrack(i + 1);
        cw -= w[i];       }
      if (cw + r > bestw)  {
        x[i] = 0;  // 搜索右子树
        backtrack(i + 1);       }
      r += w[i];
    }
```

5. 用分支限界法解装载问题时，对算法进行了一些改进，下面的程序段给出了改进部分；试说明斜线部分完成什么功能，以及这样做的原因，即采用这样的方式，算法在执行上有什么不同。

```
// 检查左儿子结点
  Type wt = Ew + w[i];   // 左儿子结点的重量
     if (wt <= c) {      // 可行结点
        if (wt > bestw) bestw = wt;
        // 加入活结点队列
        if (i < n) Q.Add(wt);
     }
  // 检查右儿子结点
     if (Ew + r > bestw && i < n)
        Q.Add(Ew);       // 可能含最优解
        Q.Delete(Ew);    // 取下一扩展结点
```

解答：斜线标识的部分完成的功能为：提前更新 bestw 值；

这样做可以尽早的进行对右子树的剪枝。具体为：算法 Maxloading 初始时将 bestw 设置为 0，直到搜索到第一个叶结点时才更新 bestw。因此在算法搜索到第一个叶子结点之前，总有 bestw=0，r>0 故 Ew+r>bestw 总是成立。也就是说，此时右子树测试不起作用。

为了使上述右子树测试尽早生效，应提早更新 bestw。又知算法最终找到的最优值是所求问题的子集树中所有可行结点相应重量的最大值。而结点所相应得重量仅在搜索进入左子树是增加，因此，可以在算法每一次进入左子树时更新 bestw 的值。

7. 最长公共子序列问题：给定 2 个序列 X={x1, x2, …, xm} 和 Y={y1, y2, …, yn}，找出 X 和 Y 的最长公共子序列。

由最长公共子序列问题的最优子结构性质建立子问题最优值的递归关系。用 c[i][j]记录序列 Xi 和 Yj 的最长公共子序列的长度。其中，Xi={x1, x2, …, xi}；Yj={y1, y2, …, yj}。当 i=0 或 j=0 时，空序列是 Xi 和 Yj 的最长公共子序列。故此时 C[i][j]=0。其它情况下，由最优子结构性质可建立递归关系如下：

$$c[i][j]=\begin{cases}0 & i=0,j=0\\ c[i-1][j-1]+1 & i,j>0;x_i=y_j\\ \max\{c[i][j-1],c[i-1][j]\} & i,j>0;x_i\neq y_j\end{cases}$$

在程序中，b[i][j]记录 C[i][j]的值是由哪一个子问题的解得到的。

（1） 请填写程序中的空格，以使函数 LCSLength 完成计算最优值的功能。

```
void LCSLength(int m, int n, char *x, char *y, int **c, int **b)
{
        int i, j;
        for (i = 1; i <= m; i++) c[i][0] = 0;
        for (i = 1; i <= n; i++) c[0][i] = 0;
        for (i = 1; i <= m; i++)
           for (j = 1; j <= n; j++) {
              if (x[i]==y[j]) {
                    c[i][j]=c[i-1][j-1]+1; b[i][j]=1;}
             else if (c[i-1][j]>=c[i][j-1]) {
                    c[i][j]=c[i-1][j]; b[i][j]=2;}
             else { c[i][j]=c[i][j-1]; b[i][j]=3; }
              }
}
```

(2) 函数 LCS 实现根据 b 的内容打印出 Xi 和 Yj 的最长公共子序列。请填写程序中的空格，以使函数 LCS 完成构造最长公共子序列的功能（请将 b[i][j]的取值与（1）中您填写的取值对应，否则视为错误）。

```
void LCS(int i, int j, char *x, int **b)
{
     if (i ==0 || j==0) return;
     if (b[i][j]== 1) {
           LCS(i-1, j-1, x, b);
           cout<<x[i];
           }
     else if (b[i][j]== 2) LCS(i-1, j, x, b);
     else LCS(i, j-1, x, b);
}
```

8. 对下面的递归算法，写出调用 f(4)的执行结果。

```
void f(int k)
{ if( k>0 )
        { printf("%d\n  ",k);
          f(k-1);
          f(k-1);
        }
}
```

基础知识

1 算法特征 ： 输入 输出 确定性 可行性 有穷性

2 描述算法的方法： 自然语言 流程图 伪代码 程序设计语言

3 欧几里德算法 ： 辗转相除法

4 常见算法的种类 ： 精确算法 启发式算法 近似算法 概率算法

5 算法复杂度 ： 运行一个算法所需的时间和空间

6 一个好的算法： 正确性 简明性 效率 最优性

7 算法的最优性： 一个算法在最坏情况下的执行时间能达到解决该类问题的时间下限

8 影响程序运行时间的要素： 程序所依赖的算法 问题规模和输入的数据 计算机系统性能

9 渐近上界记号 O：如果存在正常数 c 和自然数 $n0$，使得当 $n \geq n0$ 时有 $f(n) \leq cg(n)$，则称函

10 数 f(n)当 n 充分大时有上界，且 g(n)是它的一个上界，记做 $f(n)=O(g(n))$ 。即 f(n)的阶不高于 g(n)的阶。

11 渐近下界记号 Ω ： 如果存在正常数 c 和自然数 $n0$，使得当 $n \geq n0$ 时有 $f(n) \geq cg(n)$ ，则称函数 f(n)当 n 充分大时有下界，且 g(n)是它的一个下界，记做 $f(n)=\Omega(g(n))$ 。即 f(n)的阶不低于 g(n)的阶。

12 紧渐近界记号 Θ ：如果存在正常数 $c1,c2$ 和 $n0$，使得当 $n \geq n0$ 时有 $c1g(n) \leq f(n) \leq c2g(n)$，则称函数 f(n)与函数 g(n)同阶，记做 $f(n)=\Theta(g(n))$。即 f(n)与 g(n)的增长阶数相同。

13 多项式时间比较 $O(1)<O(\log n)<O(n)<O(n\log n)<O(n)<O(n\log n)<O(n^2)<O(n^3)$

14 指数时间算法：$O(2^n)<O(n!)<O(n^n)$

15 分治法求解的要素：1 问题能够按照某种方式分解成若干个规模较小、相互独立且（与原问题）类型相同的子问题 2 子问题足够小时可以直接求解 3 能够将子问题的解组合成原问题的解

16 分治法很自然的导致一个递归算法。

17 递归算法的时间复杂度分析:递推式 $T(n)=aT(n/b)+cn^k$，$T(1)=c$ 求解。

18（有序表）对半搜索二叉判定树的性质性质 5-1：具有 n(n>0)个内结点的对半搜索二叉判定树的左子树上有(下限)[(n-1)/2]个内结点，右子树上有[n/2]（下限）个内结点。

具有 n(n>0)个内结点的二叉判定树的高度为 [logn]+1(下限)（不计外结点）。

若 n=2h-1，则对半搜索二叉判定树是满二叉树。

若 n=2h-1，则对半搜索二叉判定树的外结点均在 h+1 层上；否则，在第 h 或 h+1 层上。（h=[logn]+1(下限)）

对半搜索算法在成功搜索的情况下，关键字值之间的比较次数不超过[logn]+1。对于不成功的搜索，算法需要进行 [logn] 或 [logn]+1 次比较。(都是下限)

对半搜索算法在搜索成功时的平均时间复杂度为 Θ(logn)。

19 两路合并排序的时间复杂度 O(nlog2n) 快速排序的时间复杂度 O（nlog2n）

20 贪心法的基本要素 1 所谓贪心选择性质是指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。2 一个问题的最优解包含其子问题的最优解，则称此问题具有最优子结构性质。

21 动态规划法的基本要素 1 最优子结构性质——用动态规划法求解的前提。当一个问题的最优解包含了其子问题的最优解时，称该问题具有最优子结构性质 2 子问题重叠性质（递归算法求解问题时）每次产生的子问题并不总是新问题，有些子问题被反复计算多次，这种性质称为子问题重叠性质。*（自底向上的求解避免重复计算）

22 回溯法 状态空间树——描述问题解空间的树形结构 1 问题状态（树中每个结点）2 解状态（候选解元组）3 答案状态（可行解元组）4 最优答案结点（目标函数取最优值的答案结点）