

姓名：\_\_\_\_\_  
 题，否则试卷无效。

一、选择题（每题2分，15题，共30分）  
 填空题答案请填写在下表中

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

1.  $T(n)$ 表示当输入规模为  $n$  时的算法效率，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.  $T(n)=3n\log_2 n$
2. 分治法的设计思想是将难以直接解决的大问题分割成规模较小的子问题，分别解决子问题，最后将子问题的解组合起来形成原问题的解，这要求原问题和子问题（ ）。
  - A. 问题规模相同，问题性质相同
  - B. 问题规模相同，问题性质不同
  - C. 问题规模不同，问题性质相同
  - D. 问题规模不同，问题性质不同

3. 回溯法在问题的解空间树中按 ( ) 策略从根节点出发。  
A. 广度优先 B. 活结点优先 C. 扩展结点优先 D. 深度优先
4. 回溯法的效率不依赖于下列 ( )。  
A. 确定解空间的时间 B. 满足显式约束的值的个数  
C. 计算约束函数的时间 D. 计算限界函数的时间
5. 下面 ( ) 是回溯法中为避免无效搜索采取的策略。  
A. 递归函数 B. 剪枝函数 C. 随机数函数 D. 搜索函数
6. 常见的两种分枝限界法为 ( )。  
A. 广度优先分枝限界法和深度优先分枝限界法  
B. 队列式分枝限界法和堆栈式分枝限界法  
C. 排列树和子集树  
D. 队列式分枝限界法和优先队列式分枝限界法
7. 下列采用最大效益优先搜索方式的算法是 ( )。  
A. 分枝限界法 B. 动态规划法 C. 贪心法 D. 回溯法
8. 下面 ( ) 是贪心算法的基本要素之一。  
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
9. 下面 ( ) 不能使用贪心算法。  
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
10. 一个问题可用动态规划法或贪心法求解的关键特征是问题的 ( )。  
A. 贪心选择性质 B. 重叠子问题  
C. 最优子结构性 D. 定义最优解
11. 备忘录法是 ( ) 的变形。  
A. 分治法 B. 回溯法 C. 贪心法 D. 动态规划法
12. 以下不属于贪心算法的是 ( )。  
A. Prim 算法 B. Kruskal 算法  
C. Dijkstra 算法 D. 广度优先遍历
13. 旅行商问题是 NP 问题吗 ( )。  
A. 是 B. 不是 C. 不一定 D. 无定论

## 第二题

得分

## 二、填空题 (每空 2 分, 10 空, 共 20 分)

答案填写在下空中

(1) \_\_\_\_\_ (2) \_\_\_\_\_ (3) \_\_\_\_\_ (4) \_\_\_\_\_

(5) \_\_\_\_\_ (6) \_\_\_\_\_ (7) \_\_\_\_\_

(8) \_\_\_\_\_ (9) \_\_\_\_\_ (10) \_\_\_\_\_

1. 算法与数据结构有联系又有 (1)。算法是思想, 数据结构则是这些思想的逻辑基础。

2.  $T(n)=O(1)$ , 当  $n=1$  时 $T(n)=2T(n-1)+1$ , 当  $n>1$  时

其时间复杂度是 (2)

3. 递归执行是通过系统的 (3) 实现的。

4. 下面是 (4) 算法设计模式。

Divide-and-conquer(P)

{ if  $|P| \leq n_0$  return adhoc(P);for( $i=1; i \leq k; i++$ )     $y_i = \text{Divide-and-conquer}(P_i);$ return merge( $y_1, y_2, \dots, y_k$ )

}

5. 蛮力法的主要缺点是设计的大多数算法的效率都不高, 主要适合问题规模 (5) 的问题求解。

6. 回溯法有 (6) 之称, 它可以系统地搜索一个问题的所有解或任意解。

7. 优先队列式分支限界法选取扩展结点的原则是 (7)。

8. 用贪心算法求解“活动安排”问题, 是按活动的 (8) 时间递增排序。



分

1. 阅读下面

```
#include <stdio.h>
#define m1(x,y) ((x)>(y)?(x):(y))
#define m2(x,y) ((x)<(y)?(x):(y))
Void M1M2(int a[],int low,int high,int &maxe,int &mine)
{
    if(low==high)
    {
        maxe=a[low];
        mine=a[low];
    }
    else if(low==high-1)
    {
        maxe=m1(a[low],a[high]);
        mine=m2(a[low],a[high]);
    }
    else
    {
        int t mid=(low+high)/2;
        int lmaxe,lmine;
        M1M2(a,low,mid,lmaxe,lmine);
        int rmaxe,rmine;
        M1M2(a,mid+1,high,rmaxe,rmine);
        maxe=m1(lmaxe,rmaxe);
        mine=m2(lmine,rmine);
    }
}

void main()
{
    int a[]={4,3,1,2,5};
    int n=sizeof(a)/sizeof(a[0]);
    int maxe,mine;
    M1M2(a,0,n-1,maxe,mine);
    Printf("m1=%d,m2=%d",maxe,mine);
}
```

4)

些思

适合问题规

解或任意解。

同递增排序。

- 求: (1) 写出这个算法的设计思想。(4分)  
(2) 写出这个算法的时间复杂度。(3分)  
(3) 写出算法实现的结果。(3分)

2. 阅读下面算法, 这是用蛮力法求数组  $a$  中大小相差最小的两个元素的差。

```
int Middif(int a[], int n)
{
    int dmin = INF;
    for(int i = 0; i <= n; i++)
        for(j = i + 1; j <= n - 1; j++)
            { int temp = abs(a[i] - a[j]);
              if (temp < dmin)
                  dmin = temp;
            }
    return dmin;
}
```

- 求: (1) 写出这个算法的时间复杂度。(3分)  
(2) 用贪心算法思想改进这个算法。(4分)  
(3) 写出改进算法的时间复杂度。(3分)

3. 一个机器人只能向下和向右移动, 每次只能移动一步, 用动态规划策略设计一个算法, 求它从  $(0, 0)$  移到  $(m, n)$  有多少条路径。

- 求: (1) 写出递推公式。(4分)  
(2) 设计算法, 给出算法的伪代码。(6分)

#### 第四题

得分

#### 四、算法实现题 (1 题, 共 20 分)

1. 装载问题, 有一批共  $n$  个集装箱要装上 2 艘载重量分别为  $c_1$  和  $c_2$  的轮船, 其中集装箱  $i$  的重量为  $w_i$ , 则采用下面的策略可得到最优装载方案。

(1) 首先将第一艘轮船尽可能装满;

(2) 将剩余的集装箱装上第二艘轮船。

求: (1) 画出装 3 个集装箱的解空间子集树。(2 分)

(2) 写出显式约束函数(选择当前集装箱)。(4 分)

(3) 写出上界函数(不选择当前集装箱)。(4 分)

(4) 算法填空

(10 分)

```
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 = r - w[i];
```

```
  if (cw + w[i] <= c)
```

```
  { x[i] = 1; // 搜索左子树
```

```
    cw += w[i];
```

```
    backtrack(i + 1);
```

```
    cw = _____;
```

```
  }
```

```
  if (_____)
```

```
  { x[i] = _____; // 搜索右子树
```

```
    backtrack(_____);
```

```
  }
```

```
  r = _____;
```

```
}
```