# 武汉大学计算机学院

# 算法设计与分析 期中测试

**姓名：________________ 学号：________________**

**学院：________________ 专业：________________**

一、请用大“O(·)”记号求下列函数的渐进表达式：$3n^2 + 10n - 1$; $n^2/10 + 2^n + 1/n$; $14 + 5/n + 1/n^2$; $\log n^2 + \sqrt{n}$; $20\log 3^n$（10 分，每小题 2 分）

解答：

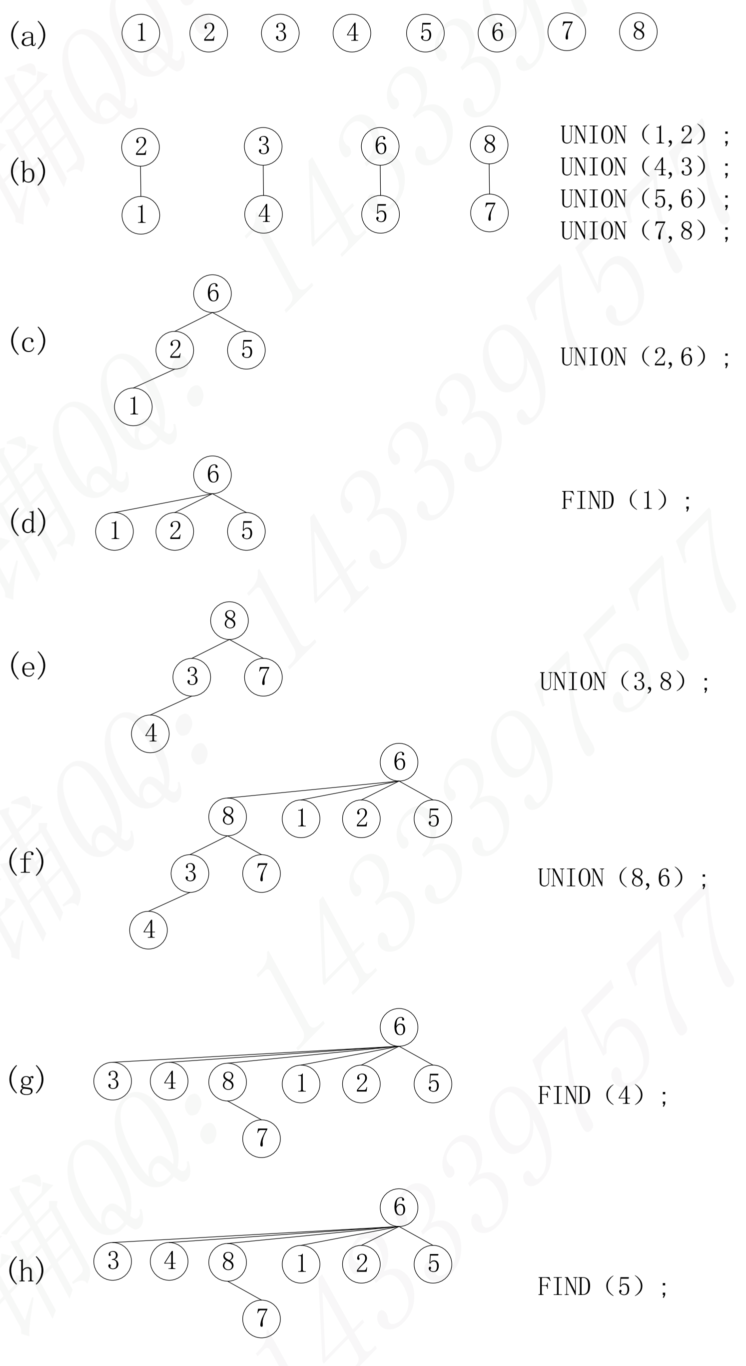
上述渐进表达式的时间复杂度分别为：

$3n^2 + 10n - 1 = O(n^2)$； $n^2/10 + 2^n + 1/n = O(2^n)$； $14 + 5/n + 1/n^2 = O(1)$；

$\log n^2 + \sqrt{n} = O(\log n)$； $20\log 3^n = O(n)$

二、 令{1},{2},{3},…,{8}是 n 个单元素集合，每个集合由一棵仅有一个结点的树表示。请用按秩合并和路径压缩措施的 UNION-FIND 算法来执行以下操作序列，并画出每一步操作完成后的树表示。（总分 20 分）合并和查找操作序列如下所示：UNION（1,2）；UNION（4,3）；UNION（5,6）；UNION（7,8）；UNION（2,6）；FIND（1）；UNION（3,8）；UNION（8,6）；FIND（4）；FIND（5）。要求：UNION 操作在同秩情况下，以后一个结点作为根结点。如 UNION（1，2），生成以 2 为根结点的树。

解答：




三、 设有 n 个小球，其中一个是劣质球，其特征是重量较轻，给你一个天平，设计一个分治算法，找出劣质球。（总分 15 分）

（1）写出算法的主要思路；（5 分）

（2）试分析算法的时间复杂度；（5 分）

（3）试分析 n=9 和 10，即 n 分别为奇数和偶数，两种情形下的分治过程。（5 分）

**解法：**（1）二对分算法思路：

①若小球个数≤2，则直接比较，找出假币。否则，转②。

②若 n%2=0，则将其分为个数相等的两部分，选择轻的部分保留，转①；否则转③。

③将 a[0…n-2]分为相等的两部分：若两部分重量相等，则 a[n-1]为劣质球，终止；若不等，则保留轻的部分，转①。

（2）以比较操作为基本运算，最好情况比较 1 次，最坏比较 logn 次，

（3）① 分成两部分：a[0…4]、a[5…9]，假定后者轻，保留 a[5…9]

② 分成三部分：a[5…6]、a[7…8]、a[9]，若前两者一样重，故劣质球为 a[9]。

四、 考试前，A 老师给同学答疑，同一时间只能给一个同学答疑，有 n 个人等待答疑，已知每个人需要答疑的时间为 ti（0<i<=n），请设计贪心算法安排排队次序，使每个人排队等候时间总和最小。（总分 15 分）

（1）请写出两种以上的贪心策略，比较它们，选出一种用于贪心算法；（5 分）

（2）写出贪心算法的主要思路；（5 分）

（3）该算法一定能够保证排队时间总和最小？请简要说明理由。（5 分）

**解法：**

（1）答疑时间短先安排；答疑时间长先安排。

（2）本题贪心算法：n 个人时间从小到大排序，就是这 n 个人最佳排队方案。求部分和的和即为所求。

（3）反证法证明：假设有最优解序列：s1,s2…sn,如 s1 不是最小的 Tmin，不妨设 sk=Tmin,将 s1 与 sk 对调，显然，对 sk 之后的人无影响，对 sk 之前的人等待都减少了，(s1-sk)>0,从而新的序列比原最优

序列好，这与假设矛盾，故 s1 为最小时间，同理可证 s2…sn 依次最小。

五、 在一个操场上一排地摆放着 N 堆石子，N 堆石子的编号为 1，2，…，N。现要将石子有次序地合并成一堆。每堆石子包含的石子个数给定，规定每次只能选相邻的 2 堆石子合并成新的一堆，并将新的一堆石子数记为该次合并的得分。（总分 20 分）

（1）假设要求计算出将 N 堆石子合并成一堆的最小得分值，已知该问题可以采用动态规划来进行求解，试写出你的动态规划算法的递归方程，并分析该递归方程能否采用递归程序来实现；（5 分）

(2) 试设计一个动态规划程序（伪代码即可），计算出将 N 堆石子合并成一堆的最小得分值；（5 分）

(3) 试分析第（2）问中你设计的动态规划算法的时间复杂度；（5 分）

(4) 如果要得到取得最小得分的合并方案，将如何修改程序，使之能够输出最优的合并方案，并分析该方法的空间复杂度（注意：最优合并方案的表示可以采用加括号的方式表示）。（5 分）

**参考答案：**注意, 本题会有多种解法, 参考答案仅仅是一种, 改卷子时一定要看清楚

（1）设 $S$(i)表示前 i 堆石子总的数量(也即价值之和)，$f$[i][j]表示把第 i 堆到第 j 堆的石头合并成一堆的最优值。则递推方程为：

$$f[i][j]=\begin{cases}\min\limits_{i\le k<j}(f[i][k]+f[k+1][j]+s[j]-s[i]) & i\neq j\\ 0 & i=j\end{cases}$$

———————————————得分 3 分

由于该递推方程递推下去包含有大量重叠子问题，所以不能直接采用递归算法来实现，递归的算法复杂度为：

$$f(n)=\sum_{k=1}^{n-1}f(k)f(\mathrm{n}-k)=\frac{1}{n}\binom{2n-2}{n-1}$$

———————————————得分 2 分

（2）算法分为初始值赋值和循环两个评分点，算法的伪代码为；

```
Algorithm dd（）
{
   for (i=1; i<=n; i++) f[i][i]=0;   //可能超出int的范围
                         -----------------------得分2分
    for (i=n-1; i>=1; i--)
        for (j=i+1; j<=n; j++)
            f[i][j]=INF;
            for (k=i; k<=j-1; k++)

f[i][j]=min(f[i][j],f[i][k]+f[k+1][j]+s[j]-s[i-1]);

-----------------------得分3分
    printf("%d\n",f[1][n]);
    return 0;
}
```

（3）算法复杂度为$\Theta(n^3)$。

-----------------------得分3分

（4）输出最优解的程序（和矩阵链相乘一样）

```
Algorithm dd(p)
 {
```

```
n ← length[p] - 1
for i ← 1 to n
  m[i, i] ← 0
end for
for l ← 2 to n
  for i ← 1 to n - (l - 1)
  j ← i + (l - 1)
  m[i, j] ← ∞
  for k ← i to j - 1
      q ← m[i, k] + m[k + 1, j] + p[i-1] p[k]p[j]
      if q < m[i, j]
      then m[i, j] ← q, s[i, j] ← k
      end if
  end for
  end for
end for
return m and s
}
```

----------------------得分 2 分

```
PRINT-OPTIMAL (s, i, j)
1 if i = j
2 then print "A"i
```

```
else print "("
PRINT-OPTIMAL (s, i, s[i, j])
PRINT-OPTIMAL (s, s[i, j] + 1, j)
print ")"
```

------------------------得分 2 分

空间复杂度为$\Theta(n^2)$。

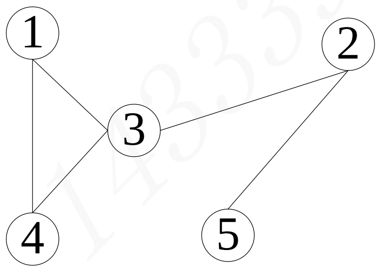
------------------------得分 1 分

六、最大团问题：给定无向图 G=(V,E)，其中 V 是非空集合，称为顶点集；E 是 V 中元素构成的无序二元组的集合，称为边集，无向图中的边均是顶点的无序对，无序对常用圆括号“( )”表示。如果给定 U⊆V，且对任意两个顶点 u，v∈U 有(u,v)∈E，则称 U 是 G 的完全子图。G 的完全子图 U 是 G 的团当且仅当 U 不包含在 G 的更大的完全子图中。G 的最大团是指 G 中所含顶点数最多的团。（总分 20 分）

（1）请设计一个回溯算法（伪代码即可）来求解最大团问题；（10 分）

（2）你设计的算法的解向量如何表示？时间复杂度是多少？（5 分）

（3）假设有如下下图所示的问题实例，




试采用你设计的算法，把求解最大团的搜索过程详细写出来。（5 分）

**参考答案：**注意，本题会有多种解法，参考答案仅仅是一种，改卷子时一定要看清楚。

（1）假设解向量采用等长的二进制编码（$x_1,x_2,\ldots,x_n$），其中 n 为图中顶点的个数，回溯算法的递归版本代码如下：

**Input**: An undirected graph G=(V,E).

**Output**: A solution vector x[1,2,…,n].

1. **for** *k*←*1* **to** *n*

2. y[k]=x*[k]*←*-1*

3. **end for**

4. *mcl*←-1

5. maxcl(1,0,n)

6. **output** y

7. **output** mcl

**Procedure** maxcl (k,r,l)

1. **x(k) = 0**

2. **if** k=n **then**

3. **if** r>mcl **then** {mcl = r, y=x} **endif**

4. **else if** r+l >macl **then** maxcl(k+1,r,l-1)

5. **endif**

6. **x(k) =1**

7. **if** 节点 k 与前面取值为 1 的节点均有边相连 **then**

8. **if** k=n **then**

9. **if** r>mcl **then** {mcl = r, y=x} **endif**

10. **else** maxcl(k+1,r+1,l-1)

11. **endif**

7. **end if**

------------------------得分 10 分

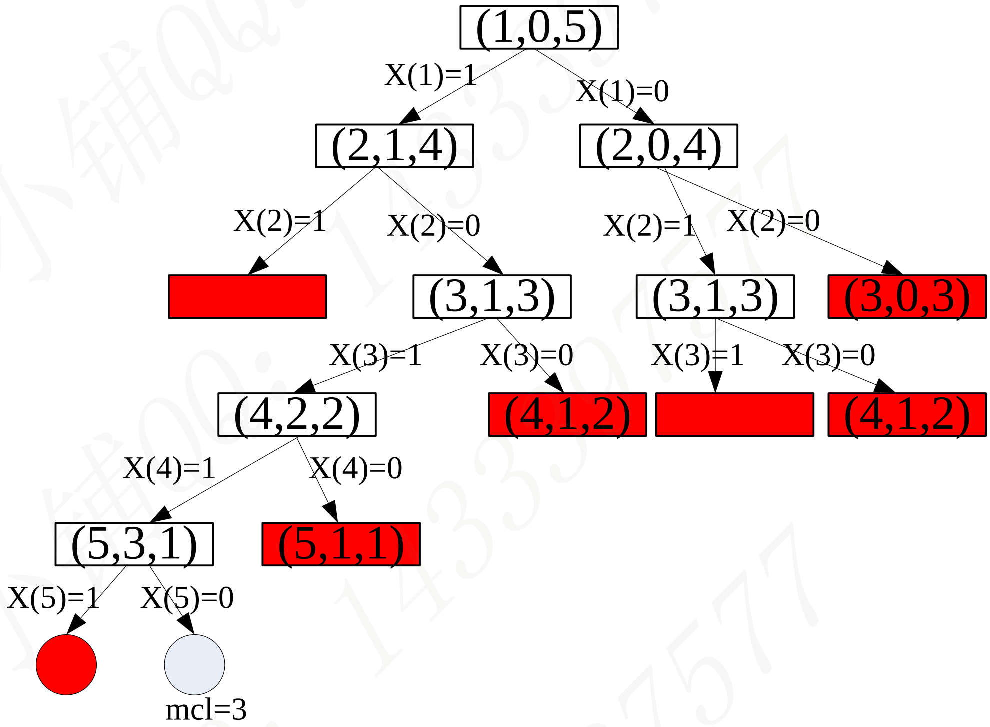
（2）解向量采用等长的二进制编码（$x_1,x_2,\ldots,x_n$），其中 n 为图中顶点的个数。

------------------------得分 2 分

时间复杂度为 $O(n2^n)$。

------------------------得分 3 分

（3）求解过程如下图所示（由于先选取 x(k)=0 的节点先生成，本实例造成的树太大，所以我们先生成 x(k)=1 的节点，不管那种做法，答案都算对），其中红色无字方框是不满足要求的中间节点，红色有字方框为被限界的中间节点，红色圆形为不满足要求的解，灰色圆形为满足要求的解。


(1,0,5)
X(1)=1
X(1)=0
(2,1,4)
(2,0,4)
X(2)=1
X(2)=0
X(2)=1
X(2)=0
(3,1,3)
(3,1,3)
(3,0,3)
X(3)=1
X(3)=0
X(3)=1
X(3)=0
(4,2,2)
(4,1,2)
(4,1,2)
X(4)=1
X(4)=0
(5,3,1)
(5,1,1)
X(5)=1
X(5)=0
mcl=3

----------------------得分 5 分