

2023 年秋算法设计与分析期末考试回忆版试题

编者：夏提雅

参与回忆者：夏提雅、Clouder、夕日坂语叶

Version: 1.2 (2023 年 12 月 1 日)

免责声明：本试题是在离开考场后，回忆出来的，不存在任何作弊行为；本试题选择题部分不保证题干、选项与原题一致，但考察的中心思想一致；本试题大题部分材料不全，希望大家踊跃补充。（考试时间：2023 年 11 月 25 日；满分：100 分；时间：120 分钟）

一、选择题（共 5 小题，每小题 2 分，共 10 分）

1. 设 $f(n) = n^2$ ， $g(n) = n^3$ ，则

A. $f(n) = O(g(n))$

B. $f(n) = \Omega(g(n))$

C. $f(n) = \Theta(g(n))$

D. $f(n) = \omega(g(n))$

2. 贪心算法与动态规划算法共同具有的性质是

A. 最优子结构性质

B. 贪心选择性

C. 重叠子问题性质

D. 分治与递归求解

3. 以下算法没有使用贪心思想的是

A. 哈夫曼编码算法

B. Prim 算法

C. Dijkstra 算法

D. 深度优先搜索算法

4. 已知一个有序表为 $\{13, 18, 24, 35, 47, 50, 62, 83, 90, 115, 134\}$ ，当用二分查找方法查找值为 90 的元素时，查找成功时，值比较的次数为

A. 2

B. 3

C. 4

D. 5

5. 下列说法不正确的是

A. 平均来说，比较排序的时间复杂度下界为 $O(n \log n)$

B. KMP 算法是线性的字符串匹配算法

C. A* 算法使用了 Best-First 的搜索策略

D. Bellman-Ford 算法是一种解决加权有向无环图中的单源最短路径的算法，其算法时间复杂度为 $O(VE^2)$

二、（10 分）

证明： $\log(n!) = \Theta(n \log n)$ （提示：证明下界时可取 $c = \frac{1}{2}$ ）

三、（10 分）

给你一个长度为 $n+1$ 的整数数组 `nums`，其中 `nums` 的所有整数都在范围 $[1, n]$ 内，`nums` 中有且仅有一个整数重复出现。现在，需要找出这个重复的数。

我们可以使用二分查找的思路解决这个问题。

算法设计思路：

定义第 i 个 `cnt` 表示 `nums` 数组中小于等于 i 的数有 `cnt` 个，那么设这个重复的数为 `target`，则对于任意的在 $[1, n]$ 中的 `num` 有

如果 $\text{num} < \text{target}$ ，那么 $\text{cnt}[\text{num}] \leq \text{num}$ ；

如果 $\text{num} \geq \text{target}$ ，那么 $\text{cnt}[\text{num}] > \text{num}$ 。

（1）根据以上思路，补全这个算法的伪代码。（8 分）

（2）给出算法的时间复杂度分析。（2 分）

`FindDuplicate(nums)`

`L` \leftarrow 1;

`R` \leftarrow `n`;

While _____ Do

`mid` \leftarrow (`L`+`R`)/ 2;

`cnt` \leftarrow 0;

For $i \leftarrow 1$ To `n` Do

If _____ Then

`cnt` \leftarrow `cnt` + 1;

If _____ Then

`L` \leftarrow `mid` + 1;

Else

`R` \leftarrow `mid` - 1;

Return `ans`;

四、（10 分）

给定一个 $m \times n$ 矩阵 `grid`，起始点为其左上角，某人从起始点出发，每次只能向下或者向右移动一步，试图达到矩阵的右下角。但矩阵中有障碍物，用 1 表示，他不能经过障碍物；矩阵中能走的位置标记为 0。求从左上角到右下角将会有多少条不同的路径。请使用动态规划算法求解该问题。

例如： `grid = [[0,0,0],`
 `[0,1,0],`
 `[0,0,0]]`

算法输出：2

- （1）写出算法的优化子结构。（不需要证明）（2 分）
- （2）写出递归方程。（4 分）
- （3）写出算法的伪代码，并分析其时间复杂度。（4 分）

五、（12 分）

给定数组 `people`。`people[i]`表示第 i 个人的体重，现在这些人需要登船，船的数量不限。每艘船最多可同时载两人，且每艘船可以承载的最大重量为 `limit`。需要求解，可以承载所有人所需的最小船数。

注意：

1. 本题需要用贪心算法求解，贪心选择性不需要证明。
2. 如果伪代码中使用了排序等，可以在伪代码中直接写明使用了某函数，该函数算法不需要另外写出。
 - （1）若 `people[i]=[1,2,3,6,3,5]`，求所需的最小船数。
 - （2）描述该问题的贪心策略。
 - （3）写出该问题的贪心算法设计思路，并据此写出求解问题的伪代码（写明注释）。

六、（12 分）

给定城市 A 到 I 之间的航班线路如下表所示，表中数字表示这两座城市之间的有向航班的用时（小时）。现在需要使用使用 A*算法求解从城市 A 出发，到达城市 I 的最短用时。

	A	B	C	D	E	F	G	H	I
A	-	2	3						
B	-	-							
C	-	-	-						
D	-	-		-					
E	-	-			-				
F	-	-				-			
G	-	-					-		
H	-	-	-	-				-	5
I	-	-	-	-					-

（1）求用邻接矩阵所表示的有向图；给出该问题 $g(n)$ 的定义；对于该问题， $h(n)$ 有以下两种定义方式，请选择其中正确的一种并说明理由。

1. $h(n) = \max\{\text{节点 } n \text{ 的所有出边的权重}\};$
2. $h(n) = \min\{\text{节点 } n \text{ 的所有出边的权重}\}.$

(2) 完成该问题使用 A* 算法求解时的搜索树填空，对于叶子节点只需写出 $g(n)$ 。

n
$h(n)$
$f(n)$
$step$

七、（12 分）

学校的体育器材室可供同学们借用体育器材，管理员需要记录器材的借出（lend）和统一归还（withdraw）操作。两种操作的示意定义如下：

lend:

```
list.append(record.time, record.equipment, record.student);
```

withdraw:

```
while(!list.empty())
```

```
list.pop();
```

（注意：每个操作的代价仅考虑对 list 表插入和删除的代价，第 i 个操作后，list 中的元素数量为 m_i ）

（1）

使用势能法分析摊还代价，势函数定义为 $\Phi(i) =$ _____
_____（1 分）

对于 lend 操作，其摊还代价 $c'_i =$ _____
_____（2 分）

对于 withdraw 操作，其摊还代价 $c'_i =$ _____
_____（2 分）

因此，对于一个有 n 次操作的操作序列，其摊还代价上界为 _____
_____（1 分）

管理系统升级后，管理员还需要增加额外操作 `count`，在记录表 `list` 内容达到 2 的整数次幂时（即 2，4，8，16...），此时如果再有一次 `lend` 操作，则管理员必须立刻在统计表 `table` 中统计每种器材被借出的数量。该操作的示意定义如下：

count:

For each record in list Do

(注意：每个操作的代价仅考虑对 **list** 表插入和删除的代价以及对 **table** 的项每一次计数的代价，第 i 个操作后，**list** 中的元素数量为 m_i)

(2)

使用势能法分析摊还代价，势函数定义为 $\Phi(i) = \frac{m_i}{2}$ ($2^t < m_i < 2^{t+1}$)

（势函数 $\Phi(i)$ 应该为分段的线性函数，你只需写出其满足此条件时的表达式）（2 分）

进行第 i 个操作 `lend` 前，若 `list` 中的元素数量 m_{i-1} 为 2 的整数次幂，其摊还代价 $c'_i =$

(1 分)

进行第 i 个操作 `lend` 前，若 `list` 中的元素数量 m_{i-1} 不为 2 的整数次幂，其摊还代价 $c'_i =$ _____

 (1 分)

对于 withdraw 操作，其摊还代价 $c'_i =$ _____

_____ (1 分)

因此，对于一个有 n 次操作的操作序列，其摊还代价上界为_____

_____ (1分)

八、（12 分）

现要给 a、b、c 3 个女生和 x、y、z 3 个男生分配座位。女生 a 允许与男生 x 或 y 坐一起；女生 b 只允许与男生 x 坐一起；女生 c 允许与男生 y 或 z 坐一起。这是一个匹配问题，请把它转化为最大流问题，然后使用 Ford-Fulkerson 算法求解。画出每一步求解的流、余图，写出其增广路，最后写出求解出的座位表。

九、（12 分）

Sunday 算法是 BMH 字符串匹配算法的一种优化算法，其思路如下：

1. 首先，将模式串 P 与主串 T 左端对齐。
2. 然后，从模式串 P 的最右端开始匹配，如果成功，则继续匹配；
3. 如果匹配失败，那么选择主串中与模式串 P 右端对齐位置的下一个字符 $T[i]$ ，在模式串 P 中从右到左依次寻找与之相等的字符。
4. 如果模式串 P 中与 $T[i]$ 相等的字符为 $P[j]$ ，那么将模式串 P 向右移动，使 $T[i]$ 与 $P[j]$ 对齐，然后重复 2 操作。
5. 如果模式串 P 中不存在与 $T[i]$ 相等的字符，那么将模式串 P 向右移动，使其左端与 $T[i]$ 的下一个字符对齐，然后重复 2 操作。

(1) 写出 Sunday 算法的偏移表计算公式。

(2) 如下表格是 Sunday 算法的匹配过程，请写每一步匹配的内容。

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
T	a	b	d	a	c	b	a	c	d	b	a	c	a	c	c	a	c
P	a	c	c	a	c												
0																	
1																	
2																	
3																	
4																	
5																	
6																	