《算法设计与分析》2018 期末考试真题

尹达恒

2020/12/21

东 南 大 学 考 试 卷（A 卷）

课程名称	算法设计与分析	考试学期	2018-2019-2	得分	
适用专业	计算机	考试形式	开卷	考试时间长度	150 分钟

（可携带纸质教材、课件、讲义、笔记）

姓名______ 学号______

线 封 密

1. 判断题（共 10 分，每小题 2 分）

   a）贪心算法是最优求解算法……………………………………………………（ ）

   b）KMP 算法为长度为m的串计算 next 数组的复杂度为$O(\log m)$…………（ ）

   c）已知一个 NP 完全问题 A 和一个 NP 问题 B，若 A 可多项式时间规约到 B 问题的一个实例，则 B 也可多项式时间规约到 A 的一个实例……………………………（ ）

   d）算法 A 的时间复杂度为$T_A = O(n\log n)$，则$T_A = O(n^2)$也正确………………（ ）

   e）动态规划算法可以多项式时间求解……………………………………………（ ）

2. 给定两个字符串$A[1,2,\ldots,m]$和$B[1,2,\ldots,n]$，请设计方法求这两个字符串的最长公共子串。（共 10 分）

3. 给定平面上n个坐标点 $P = \{p_1, p_2, \ldots, p_n\}$，请完成以下方法构建凸包。（共 10 分）

   a）如何将这n个点均分成两部分，使得两部分坐标点数相差最多为 1 个？（4 分）

   b）如何合并两部分结果得到最终结果？（4 分）

   c）算法的时间代价主要包含哪些部分？时间复杂度为多少？（2 分）

4. 有n个小朋友，每个小朋友手里拿着一张数字牌，第i个小朋友的数字牌为p_i，每张牌$p_i \le B$。小朋友i和小朋友j能匹配当且仅当$p_i + p_j \le B$。请设计贪心策略匹配最多对小朋友，分析是否可得最优解。（共 10 分）

5. 给定n个整数$a = \{a_1, a_2, \ldots, a_n\}$，以及一个整数$B$。要求在$n$个整数中找一子集，使得这些子集中整数和刚好等于$B$。请设计动态规划算法求解该问题。（共 10 分）

   a） 请写出伪多项式时间复杂度的动态规划迭代公式。（6 分）

   b） 请特别讨论初始化条件。（4 分）

6. 给定一无向图 $G = (V, E)$，每条边$e \in E$有一非负权值$\omega(e)$，每个顶点$v \in V$有一非负权值$\omega(v)$，一条顶点s到顶点t的路径上所有顶点和边的权值和是这条路径的长度。请设计方法求顶点s到顶点t的最短路径长度。（共 10 分）

7. 给定一个数组$T[1,2,\ldots,n]$，存储了第 1 天到第n天的平均气温。假设气温变化没有规律。现要求为每一天i计算一个$Warmer[i]$，记录后续气温中第一个更高气温的日期。$Warmer[i] = \min\{k | i \le k \le n, T[k] > T[i]\}$。如果不存在，则设置 $Warmer[i] = n + 1$。例如下面是一个共 7 天的例子。（共 10 分）

$i =$	1	2	3	4	5	6	7
$T[i] =$	33	24	24	32	35	29	36
$Warmer[i] =$	5	4	4	5	7	7	8

请完成下面 a 或 b 小题。

a）请设计一算法计算数组 $Warmer\,[1,2,\ldots,n]$。（6 分）

b）请设计一时间复杂度为$O(n)$的算法求解该问题。（10 分）

8. 若数组中某一元素出现次数超过一半，则称该元素为 majority 元素。给定一个整型数组$a[1,2,\ldots,n]$，请按要求设计方法寻找 majority 元素。请完成 a 或 b 小题。（共 10 分）

a）请设计一时间复杂度为$O(n\log n)$的方法寻找 majority 元素。（6 分）

b）请设计一时间复杂度为$O(n)$的方法寻找 majority 元素。（10 分）

9. 骨牌是一种游戏用具（如下所示），请解决骨牌排放搜索问题。（共 10 分）

一副骨牌共 28 张，由以下点数组合构成：

骨牌号	点数	骨牌号	点数	骨牌号	点数	骨牌号	点数
1	0 | 0	8	1 | 1	15	2 | 3	22	3 | 6
2	0 | 1	9	1 | 2	16	2 | 4	23	4 | 4
3	0 | 2	10	1 | 3	17	2 | 5	24	4 | 5
4	0 | 3	11	1 | 4	18	2 | 6	25	4 | 6
5	0 | 4	12	1 | 5	19	3 | 3	26	5 | 5
6	0 | 5	13	1 | 6	20	3 | 4	27	5 | 6
7	0 | 6	14	2 | 2	21	3 | 5	28	6 | 6

这 28 张牌可以摆成 7×8 的点数网格，其对应的骨牌号如下所示：

点数网格

6	6	2	6	5	2	4	1
1	3	2	0	1	0	3	4
1	3	2	4	6	6	5	4
1	0	4	3	2	1	1	2
5	1	3	6	0	4	5	5
5	5	4	0	2	6	0	3
6	0	5	3	4	2	0	3

骨牌号图

28	28	14	7	17	17	11	11
10	10	14	7	2	2	21	23
8	4	16	25	25	13	21	23
8	4	16	15	15	13	9	9
12	12	22	22	5	5	26	26
27	24	24	3	3	18	1	19
27	6	6	20	20	18	1	19

使用搜索算法从点数网格求对应的骨牌号图，描述状态空间的组织，有可能的话，给出相关剪枝策略。

10. 一家专卖店连续N天营业，只经营一种商品，第i天可以最多进货a_i台，进货单价p_i，可以最多将b_i台存放在仓库，单位存放费用为ω_i，必须为已签合同的客户供货c_i台，单价为s_i，以该价格该天能卖到断货。请构建一个费用网络流模型，计算利润最大的营销方案。（共 10 分）

1 题解 2

1.1 问题分析

最长公共子串问题是一个典型的动态规划问题，最长公共子序列在删去最后一个公共字符后，仍然是剩下字符串的最长公共子序列，因此可得如下递推公式：

$$C[i,j] = \begin{cases} max\{C[i-1,j], C[i,j-1]\} & (A[i] = B[j]) \\ C[i-1,j-1]+1\} & (A[i] \neq B[j]) \end{cases}$$

1.2 算法伪代码

见算法 1。

Algorithm 1: 题解 2 算法伪代码

Function *F(S)* **begin**
　Input: 字符串 A, B
　Output: 最长公共子串长度 C
　if $A[i] = B[j]$ **then** $C[1,1] = 1$ **else** $C[1,1] = 0$;
　for $i \in [2, m]$ **do**
　　for $j \in [2, n]$ **do**

$$C[i,j] = \begin{cases} max\{C[i-1,j], C[i,j-1]\} & (A[i] = B[j]) \\ C[i-1,j-1]+1\} & (A[i] \neq B[j]) \end{cases}$$

　　end
　end
　return $C[m,n]$;
end

2 题解 3

2.1 问题 a

将坐标点按横坐标排序，选出左边的 $\lfloor \frac{n}{2} \rfloor$ 个点为一部分，剩下的为第二部分。

2.2 问题 b

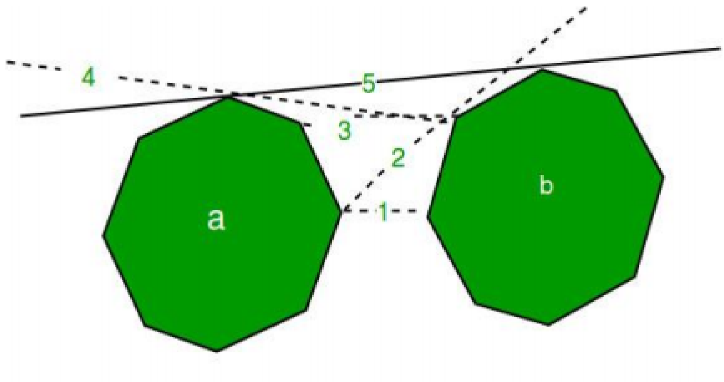


图 1: 示意图

查找合并凸包的上方连线时，从左边部分的最右点和右边部分的最左点连线开始查找：

1. 固定右边的点，遍历左凸包上在连线上方的点
2. 找到与原连线夹角最大的新连线
3. 固定左边的点，遍历右凸包上在连线上方的点
4. 找到与原连线夹角最大的新连线
5. 重复直到找不到新连线

查找合并凸包的下方连线时，从左边部分的最右点和右边部分的最左点连线开始查找：

1. 固定右边的点，遍历左凸包上在连线下方的点
2. 找到与原连线夹角最大的新连线
3. 固定左边的点，遍历右凸包上在连线下方的点
4. 找到与原连线夹角最大的新连线
5. 重复直到找不到新连线

2.3 问题 c

算法的主要时间都花在将小凸包合并为大凸包的过程上。

3 题解 4

3.1 问题分析

显然，本题的贪心选择性是：每个小朋友 p_i 都只要选择能与自己匹配的最大的 p_j 即可。

可以通过反证法证明此方法最优：如果对于某个配对 p_i, p_j，存在一个 $p_k < p_j$，使 p_i 不与 p_j 配对而与 p_k 配对时，能增加一个配对 p_j, p_l，如果要这种情况成立，那么必然有：

$$\begin{cases} p_i + p_j \leq B \\ p_k < p_j \\ p_i + p_k \leq B \\ p_k + p_l > B \\ p_j + p_l \leq B \end{cases}$$

但显然，$p_j + p_l > p_k + p_l > B$，不可能有 $p_j + p_l \leq B$，因此此方法不是最优的情况不存在。

3.2 算法伪代码

见算法 2。

4 题解 5

4.1 问题分析

此问题是典型的动态规划问题：如果存在一个子集 $b \subseteq a$，$\sum_{i \in b} i = B$，那么必有 $\sum_{i \in b, i \neq j} i = B - j$。因此有递推公式：

$$b = F(n, B) = \begin{cases} F(n-1, B - a_n) \cup \{a_n\} & F(n-1, B - a_n) \neq \nexists \\ F(n-1, B) & F(n-1, B) \neq \nexists \\ \nexists & otherwise \end{cases}$$

Algorithm 2: 题解 4 算法伪代码

```
Function F(p_i) begin
    Input: 小朋友的数字牌 {p_i}、匹配和限制 B
    Output: 能匹配的最多对小朋友 C
    对 {p_i} 进行排序;
    repeat
        选出 {p_i} 中最大的元素 p;
        从 {p_i} 中删去 p;
        if p ≥ B then continue;
        选出 {p_i} 中 ≤ B − p 最大的元素 p′;
        从 {p_i} 中删去 p′;
        C = C + 1;
    until {p_i} 中只剩 1 个或 0 个元素;
    return C
end
```

4.2 初始化条件

显然，$F(i,b)$ 在前 i 个数之和都小于 b 的情况下是必然不存在的，因此有初始化条件：

$$(\forall i \in [1,n], b \in [0,b], \sum_{i=1}^{n} a_i < b) F(i,b) = \nexists$$

和

$$F(0,b) = \emptyset$$

5 题解 6

5.1 问题分析

显然，这个问题只不过是最短路径问题的一个变型，在计算路径长度时加上点产生的代价即可。

5.2 算法伪代码

见算法 3。

Algorithm 3: 题解 6 算法伪代码

Function $F(E, V)$ **begin**

  Input: 邻接矩阵 E，顶点权值 V，起点 i，终点 j

  Output: 起点到终点的最短距离 s

  记录起点 i 到每个顶点距离 S，$S[k] = E[i, k], k \in [1, n]$;

  记录未找到最短路径的点集合 $E' = E$，将点 i 从 E' 中去除;

  初始化一个队列 Q;

  repeat

    找到 E' 中 $S[k]$ 最小的点 k;

    将点 k 从 E' 中去除;

    for $l \in E'$ **do**

      if $S[k] + E[k, l] < S[l]$ **then**

        $S[l] = S[k] + E[k, l]$;

      end

    end

  until $E' = \emptyset$;

  return $S[j]$

end

---

6 题解 7

6.1 问题分析

对于某一天的气温 $T[i]$ 后的气温要么比它高，要么比他低。

- 如过 $T[i]$ 后的某天比它低：需要继续向前查找
- 如过 $T[i]$ 后的某天比它高：$T[i]$ 到该天之间的全部 $Warmer$ 值都等于该天温度

6.2 算法伪代码

见算法 4。

Algorithm 4: 题解 7 算法伪代码

Function T **begin**
    Input: 温度列表 T
    Output: 后续气温中第一个比 $T[i]$ 高的温度 $Warmer[i]$
    初始化一个栈 S，$T[1]$ 入栈;
    $k = 1$;
    for $i \in [2, n]$ **do**
        if $T[i] > T[S_{top}]$ **then**
            repeat
                $Warmer[S_{top}] = T[i]$;
                S 出栈;
            until $T[i] \leq T[S_{top}]$ 或栈 S 为空;
        end
        i 入栈 S;
    end
    repeat
        $Warmer[S_{top}] = n + 1$;
        S 出栈;
    until 栈 S 为空;
    return $Warmer$
end

7 题解 8

7.1 问题分析

显然，可以依次扫描数组元素，给每种元素进行计数，如果和当前元素相同则加一，否则减一，如果计数值小于 0 了，那就将计数值置 0 并更换当前元素。如果一个元素在数组中出现的次数超过一半，那么数组中必然存在一个界限，从此处开始的元素计数值不会降到 0，该值即为所求的 majority。

7.2 算法伪代码

略。

8 题解 9

8.1 问题分析

对于某一天的气温 $T[i]$ 后的气温要么比它高，要么比他低。

- 如过 $T[i]$ 后的某天比它低：需要继续向前查找
- 如过 $T[i]$ 后的某天比它高：$T[i]$ 到该天之间的全部 $Warmer$ 值都等于该天温度

8.2 算法伪代码

见算法 5。

Algorithm 5: 题解 9 算法伪代码

Function T **begin**

  Input: 温度列表 T

  Output: 后续气温中第一个比 $T[i]$ 高的温度 $Warmer[i]$

  初始化一个栈 S，$T[1]$ 入栈;

  $k = 1$;

  for $i \in [2, n]$ **do**

    if $T[i] > T[S_{top}]$ **then**

      repeat

        $Warmer[S_{top}] = T[i]$;

        S 出栈;

      until $T[i] \leq T[S_{top}]$ 或栈 S 为空;

    end

    i 入栈 S;

  end

  repeat

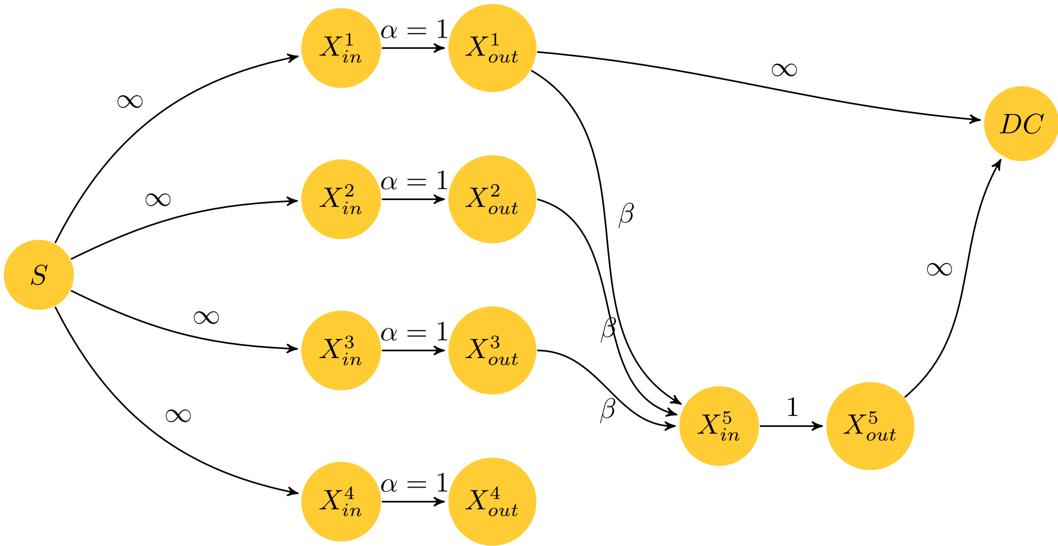
    $Warmer[S_{top}] = n + 1$;

    S 出栈;

  until 栈 S 为空;

  return $Warmer$

end



X_{in}^1
$\alpha = 1$
X_{out}^1
X_{in}^2
$\alpha = 1$
X_{out}^2
X_{in}^3
$\alpha = 1$
X_{out}^3
X_{in}^4
$\alpha = 1$
X_{out}^4
S
∞
∞
∞
∞
∞
β
β
β
X_{in}^5
1
X_{out}^5
∞
DC