

acm答案

一、选择题

1. D
2. C
3. C
4. D
5. B

二、填空题

1. 确定性、有穷性
2. 计算中间位置 $mid=\lfloor (low+high)/2 \rfloor$, 将查找范围缩小为 $a[low:mid]$ 或 $a[mid+1:high]$
3. $O(n^3)$
4. n-7元组；子集树、排列树、混合树
5. 队列式分支限界法、优先队列式分支限界法
6. 数值随机化算法、拉斯维加斯算法、舍伍德算法
7. 引入松弛变量
8. 0

三、简答题

1. 特点：通过随机选择推进计算，可能找不到解，但找到的解一定正确。举例：八皇后问题中随机放置皇后，满足条件即得解。
2. 引入松弛变量。
3. 按字符频率构建哈夫曼树，频率高的字符编码短，频率低的编码长。
4. 最优子结构和重叠子问题。
5. 特点：利用随机性、得17到近似解、结果随计算量增加收敛。举例：通过在正方形内随机投点计算圆周率 π 的值。

四、算法应用题

1. 合并排序过程

- 首次划分：将序列 {3,1,9,23,12,6} 划分为 {3,1,9} 和 {23,12,6}。
- 递归排序：{3,1,9} 排序为 {1,3,9}, {23,12,6} 排序为 {6,12,23}。
- 合并结果：合并 {1,3,9} 和 {6,12,23}, 得到最终排序 {1,3,6,9,12,23}。

2. Kruskal 算法求解最小生成树

- 按边权从小到大排序边：(1,3,1)、(4,6,2)、(3,4,4)、(2,3,5)、(1,5,5)、(1,2,6)、(2,5,6)、(3,5,6)、(1,4,5)、(5,6,6)。
- 依次选边（不形成环）：
 1. 选 (1,3), 连接顶点 1、3。
 2. 选 (4,6), 连接顶点 4、6。
 3. 选 (3,4), 连接顶点 1、3、4、6。
 4. 选 (2,3), 连接顶点 2。
 5. 选 (1,5), 连接顶点 5。
- 最终最小生成树包含边：(1,3)、(4,6)、(3,4)、(2,3)、(1,5)。

3. 0-1 背包问题 (动态规划跳跃点算法)

$$\circ \text{ 公式: } V(i, j) = \begin{cases} V(i-1, j) & (w_i > j) \\ \max\{V(i-1, j), V(i-1, j-w_i) + p_i\} & (w_i \leq j) \end{cases}$$

◦ 求解过程 (简要) :

◦ 初始化: $V(0, j) = 0$, $V(i, 0) = 0$ 。

3. ◦ ■ ◦ $i = 1$ ($w = 2, p = 18$) : $j \geq 2$ 时 $V(1, j) = 18$.

◦ $i = 2$ ($w = 3, p = 15$) : $j = 5$ 时 $V(2, 5) = \max\{18, 0 + 15\} = 18$, $j = 5$ 时 更新为 $18 + 15 = 33$.

◦ $i = 3$ ($w = 4, p = 8$) : $j = 9$ 时 $V(3, 9) = \max\{33, V(2, 5) + 8 = 41\}$.

◦ $i = 4$ ($w = 4, p = 4$) : $j = 12$ 时 $V(4, 12) = \max\{41, V(3, 8) + 4 = 41\}$.

◦ 最优值为 41.

4. 增广路算法求最大流

◦ 标号与增广过程

(简要) :

1. 源点 1 标号 $(1, +\infty)$, 检查邻边:

- $(1,2)$ 剩余容量 3 → 顶点 2 标号 $(1,3)$ 。
- $(1,3)$ 剩余容量 3 → 顶点 3 标号 $(1,3)$ 。

2. 顶点 2 检查邻边:

- $(2,4)$ 剩余容量 4 → 顶点 4 标号 $(2,3)$ 。
- $(2,5)$ 剩余容量 1 → 顶点 5 标号 $(2,1)$ 。

3. 顶点 4 检查邻边 $(4,6)$ 剩余容量 3 → 顶点 6 标号 $(4,3)$, 找到增广路 1-2-4-6, 增流 3, 更新边流量:

- $(1,2)=(3,3)$, $(2,4)=(4,3)$, $(4,6)=(3,3)$ 。

4. 重复标号过程, 继续寻找增广路 (如 1-3-5-6 等), 最终最大流为 6 (具体步骤需结合图示标注)。

五、算法设计 (n 后问题)

算法策略: 回溯法

1. 思想: 按深度优先搜索解空间树, 逐行放置皇后, 检查是否满足不冲突约束 (同行、同列、同对角线无其他皇后), 不满足则回溯。

2. 主要步骤:

- 初始化 $n \times n$ 棋盘, 全为 0 (0 表示空, 1 表示皇后)。
- 定义冲突检查函数: 判断 (i,j) 位置是否与已放皇后的行、列、对角线冲突。
- 递归放置皇后: 从第 1 行开始, 对每行每列尝试放置, 若安全则放皇后并递归下一行; 若所有列无解则回溯上一行调整。
- 当第 n 行放置成功时, 输出棋盘作为一种方案。

示例方案 ($n=4$) :

```
0 1 0 0
0 0 0 1
1 0 0 0
0 0 1 0
```

