

1. 算法设计题：【本题 15 分】

一个  $n$  元数组  $A[1 \cdots n]$ ，如果某个元素的数量超过  $n/2$  个，称其为数组的主元素。假设比较两个元素大小的时间不是常数，但判定两个元素是否相等的时间是常数，要求对于给定数组  $A$ ，设计算法判定其是否有主元素，如果有，找到该元素。请采用分治法，设计时间复杂性不大于  $O(n \log n)$  的算法。

请先简述算法思路【5 分】，再写核心伪代码【8 分】，最后分析算法复杂度【2 分】。

2. 算法应用题：动态规划法求解矩阵链连乘问题【本题共 15 分】

(1) 已知 5 个矩阵的维数分别为  $A_1: 3 \times 15$ ,  $A_2: 15 \times 10$ ,  $A_3: 10 \times 5$ ,  $A_4: 5 \times 10$ ,  $A_5: 10 \times 20$ 。请用动态规划法求  $A[1, 5]$  最小乘法次数。请给出递推式、 $M$  矩阵（记录最小乘法次数）、 $S$  矩阵（记录最佳断点位置），并给出最终的最优加括号方式。【8 分】

(2) 请画出该最佳连乘方式所对应的语法树。【3 分】

(3) 已知矩阵链连乘与凸多边形最优三角剖分是同构问题，请画出上述矩阵连乘及最佳连乘方式所对应的凸 6 边形  $\{v_0, v_1, \dots, v_5\}$  的最优三角剖分，以及该三角剖分的语法树（请在凸多边形内直接画），并指出凸多边形的边与弦分别对应矩阵链连乘的哪个部分。【4 分】

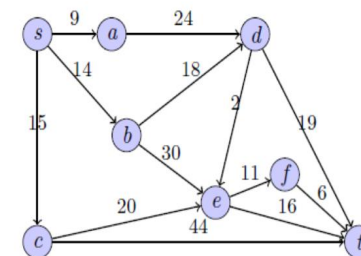
3. 算法应用与对比【本题共 12 分】。已知由 8 个整数构成的序列：(12, -4, 7, -5, 16, -4, 8, 7)。

(1) 请用分治法求最大子段和。【请给出分解与合并的具体步骤及最后的子段及子段和，4 分】

(2) 请用动态规划法求最大子段和。【请给出递推式、计算过程及最后的子段及子段和，4 分】

(3) 请根据该题，对比分治法与动态规划法的异同。【请给出共不少于 4 个异同点，4 分】

4. 算法应用与对比【本题共 18 分】。已知有向图如右侧所示：



(1) 请用 Dijkstra 单源最短路径算法求解从  $s$  到其它 7 个顶点的最短路径。【请列表计算并给出最终最短路径及长度，7 分】。

(2) 若去掉方向，将该图改为无向图，请用 Prim 算法求解最小生成树并给出其边权之和。【3 分】

(3) 请根据该例，对比 Dijkstra 和 Prim 算法的异同。【请给出共不少于 4 个异同点，4 分】

(4) 已知动态规划法又称“填表法”，Dijkstra 单源最短路径算法也填了表格，请问 Dijkstra 算法是动态规划算法吗？为什么？【4 分】

5. 算法应用题：利用分支限界法求解背包问题。【本题共 10 分】

(1) 已知一个固定容量的背包  $v=10$  和 4 个物品，每个物品有其重量和价值，重量分别为 (2, 3, 4, 7)，价值分别为 (1, 3, 5, 9)。如果每个物品只能拿 1 个或不拿，请利用优先队列式分支限界法求不超过背包容量的前提下可获得的最大价值，设计剪枝函数，画出解空间搜索树，写出最优解。【5 分】

(2) 已知一个固定容量的背包  $v=13$  和 4 个物品。每个物品有其重量和价值，重量分别为 (2, 3, 4, 7)，价值分别为 (1, 3, 5, 9)。如果每个物品可以拿 2 个、1 个或不拿，利用队列式分支限界法求不超过背包容量的前提下可获得的最大价值，设计剪枝函数，画出解空间搜索树，写出最优解。【5 分】

6. 算法设计题：【本题 20 分】。给定  $n$  个正整数  $W_i$  和一个正整数  $m$ ，在这  $n$  个正整数中找出一个子集，使得子集中的正整数之和等于  $m$ 。

(1) 请设计一个回溯算法求解该问题。要求写出算法思想【3 分】，设计剪枝函数【4 分】，写出核心代码【6 分】。

(2) 有 6 个正整数 (2, 4, 9, 7, 5, 1)，要求利用上述算法求出子集的正整数之和为 12。请画出解空间搜索树【5 分】，写出最优解【2 分】。

7. 合并排序算法。【本题共 10 分】

```
public static void mergeSort(Comparable a[], int left, int right)
{
    if (left < right) {
        //至少有 2 个元素
        int i = (left + right) / 2;
        //取中点
        ①;
        //处理左子段
        ②;
        //处理右子段
        merge(a, b, left, i, right);
        //合并到数组 b
        copy(a, b, left, right);
        //复制回数组 a
    }
}
```

(1) 请填写程序中空缺部分；【4 分】；(2) 写出该算法的时间复杂度的递归方程和求解过程。【6 分】