

算法分析与设计1

1 P、NP、NPC问题

P问题：存在能够在多项式时间内解决的算法。

NP问题：随机生成一个结果，存在能够在多项式时间内判断这个结果是否满足要求的算法。

NPC问题：该问题是NP问题，并且，对于任何NP问题，能够在多项式时间内将其规约为该问题。

补充：P、NP、NPC问题都属于判定问题，即使是求值，也能够转化为判定问题。

关系：

- P问题就是NP问题（因为既然能够在多项式时间内求解，就必然能够使用多项式时间内判定）
- NPC问题就是NP问题（本身NPC的定义就是基于NP）

2 比较两个函数的增长

使用洛必达法则比较两个函数的增长

- $O(n)$ 代表上界
- $\Omega(n)$ 表示下界
- $\theta(n)$ 表示时间复杂度相等

3 测试一内容

1. 求解递归方程；写出时间复杂度符合该递推关系的算法

$$T(n) = 2T(n/2) + n; T(1) = 0$$

$$A(n) = A(n-1) + n - 1; A(0) = 0$$

2. 请阅读代码并回答问题

1. 指出在进行时间性能分析时需要统计的基本操作；
2. 写出最好情形下的关于基本操作次数的递推关系式及其初值，并进行求解，并判定时间复杂度属于哪个量阶；
3. 写出最坏情形下的关于基本操作次数的递推关系式及其初值，并进行求解，并判定时间复杂度属于哪个量阶；

```

function binay_search(arr, low, high, x):
    if high >= low:
        mid = (high + low) / 2;
        if arr[mid] == x:
            return mid
        else if arr[mid] > x:
            return binay_search(arr, low, mid-1, x)
        else:
            return binay_search(arr, mid+1, high, x)

```

3. 什么是任务分配问题；请使用文字描述如何用蛮力法求解该问题；同时需要分析算法的时间复杂度。

4 测试二内容

1. 假设你有一个背包，它的最大承重量是6磅，现在你有5个物品可选择放入背包中：

物品A，重量1磅，价值¥15

物品B，重量3磅，价值¥20

物品C，重量4磅，价值¥30

物品D，重量5磅，价值¥55

物品E，重量2磅，价值¥65

目标是选择一些物品放入背包中，使得总重量不超过背包的最大承重量，并使选择物品的总价值最大。

1. 请定义 $f(n, W)$ 的含义，并写出关于 $f(n, W)$ 的递归公式（包括边界值）。

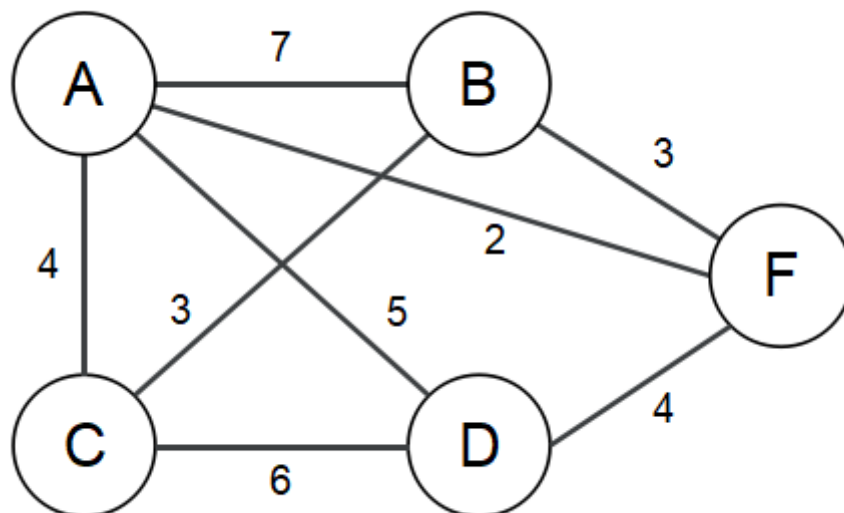
2. 填表进行求解。

2. 应用分支界限法求解旅行商问题，要求：

1. 定义下界的估计式或说明。

2. 画出以状态空间树（作图以A位根节点），并给出每个状态的下界估计。

3. 给出销售商的环游路径以及最优解。



5 最小生成树

- Prim算法：最小生成树。每次从“可延展边”中选择最小边。
- Kruskal算法：最小生成树。每次“全局”选择的最小边，检查最小边的两个点是否联通，不联通则放入。（需要使用并查集）

6 六个分类的经典问题

查看《算法分析与设计2》