222019321062074 冯春霖

为精简算法，忽略所有输入输出函数，所有输入内容已在程序中给定

算法分析题 5-3 01 背包问题（回溯算法）：

```
// n 为物品个数，c 为背包容量，obj 为物品列表，obj[i]表示第 i 个物品，obj[i][0]表示第 i 个物品的重量，obj[i][1]表示第 i 个物品的价值
const package_01 = (n, c, obj) => {
    // res 记录最大的总价值
    let res = 0
    // bt(num)表示回溯遍历到第 pos 个物品
    const bt = (num, weight, value) => {
        // 遍历完全部物品时，递归到达出口，记录结果
        if (num === n) {
            res = Math.max(res, value)
            return
        }
        // 如果该物品可选，尝试选该物品
        if (weight + obj[num][0] <= c) {
            bt(num + 1, weight + obj[num][0], value + obj[num][1])
        }
        // 不选该物品
        bt(num + 1, weight, value)
    }

    bt(0, 0, 0)
    // 算法结束，返回结果
    return res
}

// 测试数据：共五件物品如 obj 所示，背包容量为 12
const n = 5, c = 12
const obj = [[2, 4], [3, 5], [4, 7], [5, 6], [3, 6]]
// 使用测试数据进行计算，输出结果
console.log(package_01(n, c, obj)) // 输出结果为 22，选择了第 1235 项物品
```

本题使用回溯算法遍历决策树。决策树的每个分支为当前状态下的选择列表，即选或不选当前物品，已经过的路径为已做出的选择，即当前已选的物品的重量和价值，结束条件即已遍历所有物品，无法再继续，此时应更新结果。

回溯算法的重点就是要明确选择列表、路径和结束条件，在进入下一层时要及时更新路径，返回上一层时要将路径回退。

算法实现题 5-4 硬运动员最佳配对问题

```
// n, P, Q 含义如题所示
const match = (n, P, Q) => {
```

```
    // res 记录最大结果
    let res = 0
     //  track 记录已选择的运动员组合，track[i][0]表示第 i 组的男运动员，
track[i][1]表示第[i]组的女运动员
    let track = []
    // bt(s, t)表示尝试第 i 位男运动员和第 j 位女运动员后可能的搭档
    const bt = (s, t) => {
        // 已组成 n 组搭档后计算当前配对方式的竞赛优势并更新结果
        if (track.length === n) {
            let currRes = 0
            track.forEach(e => currRes += P[e[0]][e[1]] * Q[e[1]][e[0]])
            res = Math.max(res, currRes)
            return
        }
        // 回溯
        for (let i = s; i < n; i++) {
            for (let j = t; j < n; j++) {
                // 分别尝试每一个可能的组合
                track.push([i, j])
                bt(i + 1, j + 1)
                // 回退
                track.pop()
            }
        }
    }
    // 进行回溯并返回结果
    bt(0, 0)
    return res
}

// 使用题设数据进行测试并输出结果
const n = 3
const P = [[10, 2, 3], [2, 3, 4], [3, 4, 5]]
const Q = [[2, 2, 2], [3, 5, 3], [4, 5, 1]]
console.log(match(n, P, Q)) // 输出结果为 52
```

本题为求解排列问题的同类题，决策树构成为：路径：已选择的运动员配对；选择列表：可被选择的运动员；结束条件：所有运动员均已配对。
在到达结束条件时计算并更新结果，对于决策树的每一层，已配对的运动员组数是相等的。

实验心得：
回溯算法常被用来解决排列问题、组合问题，排列问题和组合问题的不同点在于排列问题的元素不能被多次选择，01 背包和运动员配对问题都属于典型的排列问题，而完全背包属于组合问题，在解决排列问题和组合问题时要通过不同的结束条件和选择列表进行约束，以得到正确答案