# 算法分析与设计第四次实验

姓名：陈光坤 学号：202208010525 班级：计科2205班

## 1.运动员最佳配对问题

1.1实验目的

本实验的目的是使用回溯算法求解一类最优匹配问题，具体来说，就是在一个特定场景下的男女运动员配对问题。给定一组男运动员和女运动员，计算在所有可能的匹配方案中，哪一种方案的配对优势最大。

1.2实验原理

每次选择一个男运动员和一个女运动员进行配对，直到所有男运动员都有配对。

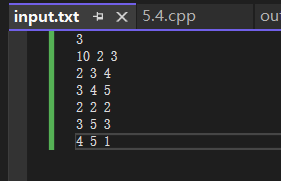
对每一个有效配对，计算当前配对方案的总优势，在遍历所有配对方案后，选择具有最大优势的方案。在搜索过程中，当发现某个配对无就可以及时剪枝，跳过无效的分支，避免不必要的计算。

1.3实验步骤

第一行包含整数 N，表示男运动员和女运动员的数量。

接下来是 N \* N 个整数，第一组 N \* N 个整数是男运动员之间的竞赛优势矩阵，第二组是女运动员之间的竞赛优势矩阵。

使用这些数据来填充 maleAdvantage 和 femaleAdvantage 矩阵。

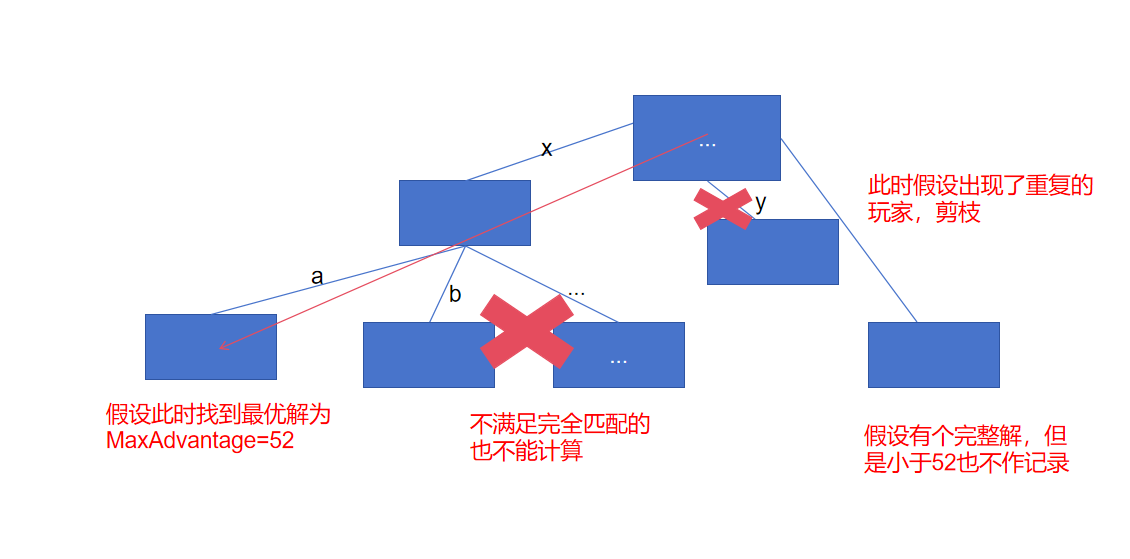


MaxAdvantage 用于记录最优的匹配优势。

currentMatch 数组表示当前的配对状态，初始值为0，表示没有运动员被配对。

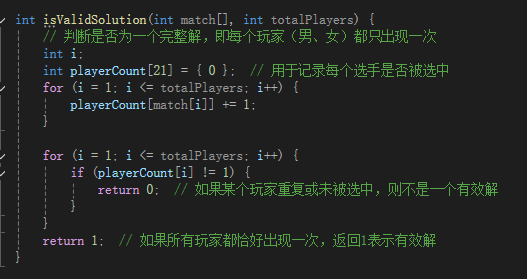
player 变量用来遍历所有的男运动员，初始值为1，表示从第一个男运动员开始。

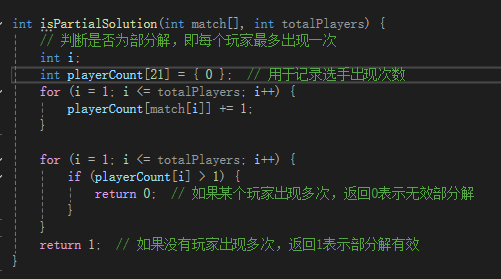
深度按男运动员编号走，每次与所有女运动员尝试匹配

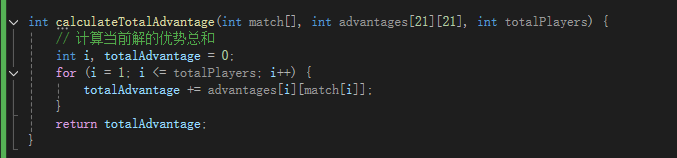


最终，输出最优解 maxAdvantage，即最大配对优势。

1.4关键代码

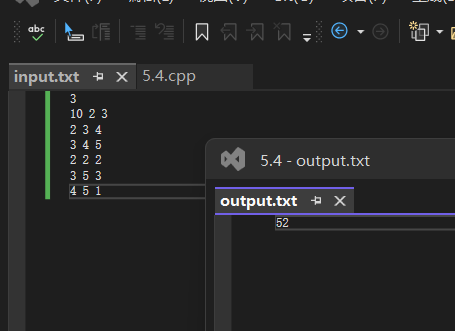








1.5测试结果



每次回溯需要尝试 N 次选择（每个男运动员配对 N 个女运动员），因此回溯的总次数是 N!N!N!（即男运动员的排列数）。这是因为每个男运动员都可以与不同的女运动员配对，回溯的过程会遍历所有的排列情况。

时间复杂度为：O(N!)

回溯使用的递归栈深度最多为 N，因为我们需要为每个男运动员递归地选择配对，因此递归的最大深度为 N。每次递归调用占用的空间为常数级别，递归深度为 N，所以递归栈的空间复杂度为 O(N)

1.6实验心得

回溯算法在解决某些组合优化问题时，提供了一种简单而直接的求解方式。但回溯的暴力搜索方法往往面临高时间复杂度的问题，因此优化和剪枝策略是实现高效求解的关键。通过此次实验，我不仅加强了对回溯算法的理解，还体会到了如何在实际问题中运用剪枝策略来提升算法的性能。我将来会继续深入学习更多的优化技术，例如动态规划、分治法等，以解决更加复杂的算法问题。

## 2.无向图的最大割问题

2.1实验目的

学习和理解分支限界算法的基本原理、工作方式及其在图论中的应用，特别是用于求解割集问题。通过使用分支限界法优化搜索空间，学习如何通过合理的剪枝策略提高算法的效率。

2.2实验原理

处理的是图的割集问题，目标是找到一个割集，使得割集的边数最大。通过优先队列维护当前最优的割集，并且在扩展节点时，只有当前割集的边数加上剩余的边数大于当前最优解时，才继续扩展。树的每个节点表示一个割集的状态，每个状态扩展成左子节点（加入当前顶点）和右子节点（不加入当前顶点），并根据剪枝条件决定是否继续深入。

2.3实验步骤

使用二维数组 G[MAX][MAX] 存储图的邻接矩阵，记录各个顶点之间是否有边连接。

结点类用于表示搜索树中的一个节点，包含当前层（dep）、割边数量（cut）、剩余边数量（e）以及一个解向量（x，表示哪些顶点被包含在割集里）。

重载了 < 运算符，使得优先队列中的节点按割边数量（cut）降序排列，即割边多的优先被扩展。

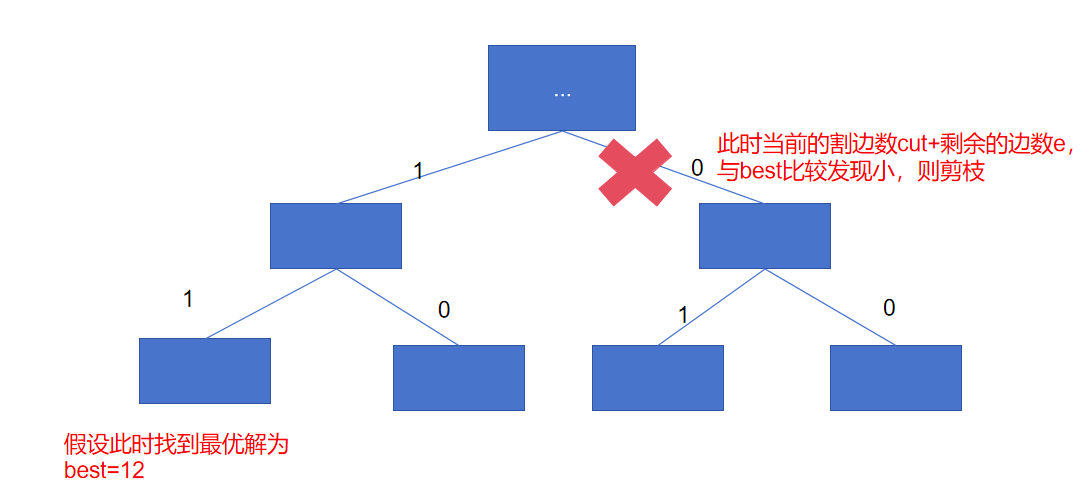
对于左子节点，表示当前顶点加入割集中；对于右子节点，表示当前顶点不加入割集。

1.初始化根节点，开始从第一个顶点开始进行搜索。

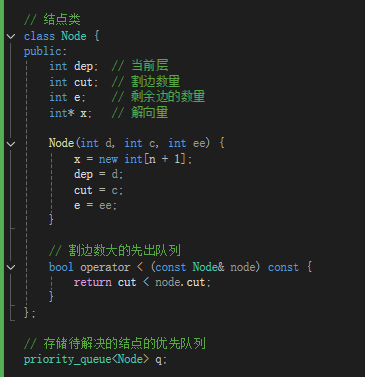
2.使用优先队列 q 存储当前待处理的节点，每次从队列中取出一个节点进行扩展。

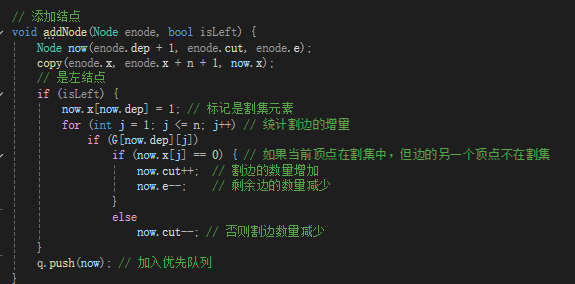
3.对于每个节点，分别尝试加入当前顶点（左子节点）和不加入当前顶点（右子节点），并根据当前割边数量和剩余边数量来决定是否继续扩展。

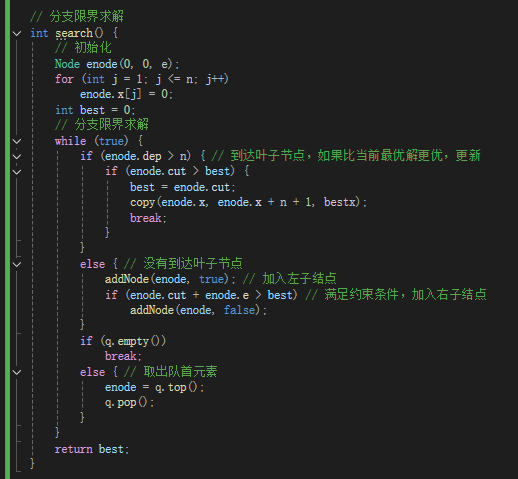
4.如果当前节点达到叶子节点（即处理完所有顶点），检查是否找到更优的解。



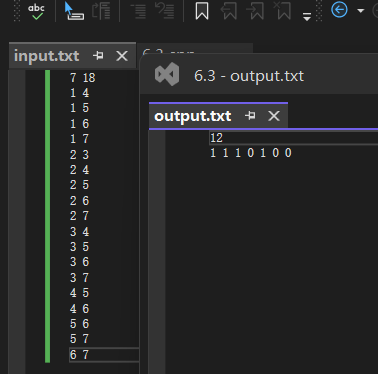
2.4关键代码







2.5测试结果



在最坏情况下，算法需要遍历所有可能的子集，并且每次操作都需要 O(n) 的时间。最坏情况下，优先队列中会有 O(2^n) 个节点。因此，优先队列的时间复杂度为O(n \* 2^n)

由于使用优先队列来存储解空间中的所有节点，以及存储图的信息。优先队列的空间复杂度为O(n \* 2^n)

2.6实验心得

通过本次实验，我不仅掌握了分支限界法的基本思想，还加深了对图论问题、图的表示方法和优先队列等数据结构的理解。在实际编程过程中，剪枝技术和优先队列的有效使用大大提高了算法的效率，但对于大规模问题，仍然面临时间和空间复杂度的挑战。