

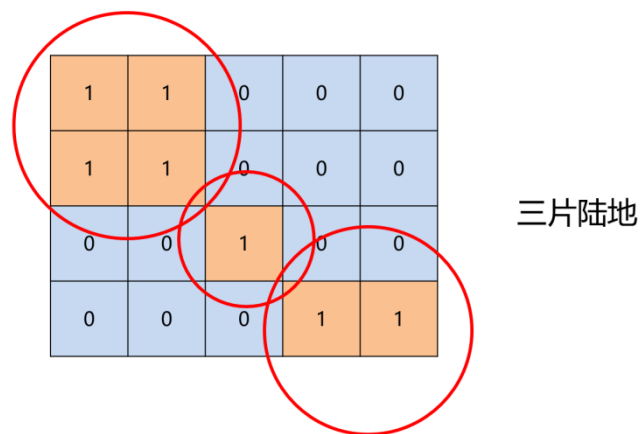
实验二 智能搜索技术

一. DFS & BFS

1. 问题描述

给你一个由 '1'（陆地）和 '0'（水）组成的二维网格，请你计算网格中岛屿的数量。岛屿总是被水包围，并且每座岛屿只能由水平方向和/或竖直方向上相邻的陆地连接形成。此外，你可以假设该网格的四条边均被水包围。

要求：一题两解，即分别使用 DFS 与 BFS 对该问题进行解答。



2. 样例

输入样例

输入给定的“地图”

```
grid = [  
    ["1","1","0","0","0"],  
    ["1","1","0","0","0"],  
    ["0","0","1","0","0"],  
    ["0","0","0","1","1"]  
]
```

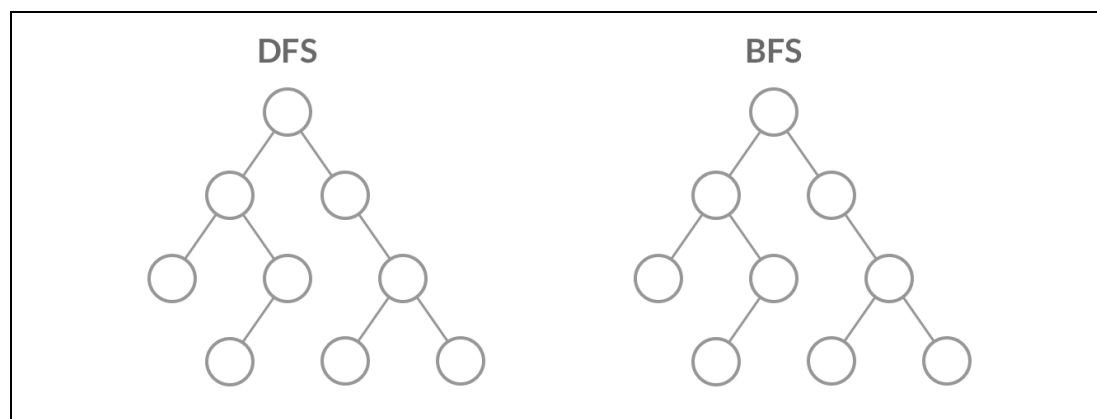
输出样例

地图上陆地的个数。

3

3. 实验分析

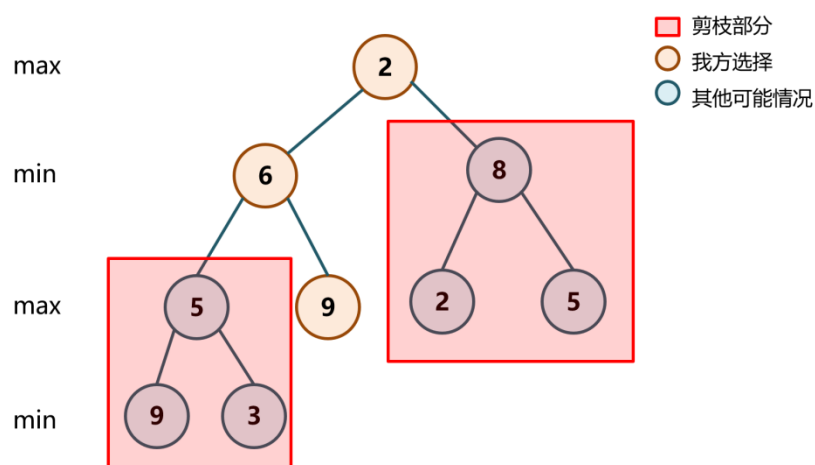
DFS 与 BFS 均为最简单也是最基础的搜索方法，无论是考研还是去找工作，该问题均会被考察。该问题与求解方法较为简单，不再对问题进行分析。



二. α - β 剪枝技术

1. 问题描述

α - β 剪枝这一概念最早来自于人机博弈，是 min-max 算法的改进版。在人机博弈中，双方回合制地进行走棋，己方考虑当自己在所有可行的走法中作出某一特定选择后，对方可能会采取的走法，从而选择最有利于自己的走法。这种对弈过程就构成了一颗博弈树，双方在博弈树中不断搜索，选择对自己最为有利的子结点走棋。在搜索的过程中，将取极大值的一方称为 max，取极小值的一方称为 min。max 总是会选择价值最大的子结点走棋，而 min 则相反。这就是极小化极大算法的核心思想。



样例

输入：为了简化操作，我们直接构造完全二叉树作为博弈树，这里按照层序遍历给出一棵树的结点。（见上图示例）

2 6 8 5 9 2 6 9 3

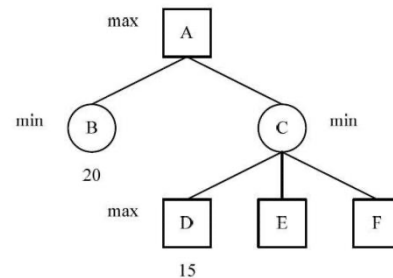
输出：被剪枝的最顶层的结点

8 5

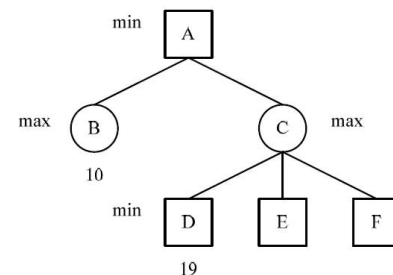
2. 问题分析

极小化极大算法最大的缺点就是会造成数据冗余，而这种冗余有两种情况：①极大值冗余；②极小值冗余。相对应地， α 剪枝用来解决极大值冗余问题， β 剪枝则用来解决极小值冗余问题，这就构成了完整的 α - β 剪枝算法。接下来我们对极大极小值冗余和具体剪枝过程作简要介绍。

α 剪枝 结点下的数据为该结点的值，结点 B 的值为 20，结点 D 的值为 15，这里，C 为取极小值的 min 结点，因此结点 C 的值将小于等于 15；而结点 A 为取最大值 max 的结点，因此 A 只可能取到 B 的值，也就是说不再需要搜索 C 的其他子结点 E 和 F 的值就可以得出结点 A 的值。这样将结点 D 的后继兄弟结点减去称为 α 剪枝。



β 剪枝 结点 B 的值为 10，结点 D 的值为 19，这里，C 结点为取最大值 max 结点。因此，C 的值将大于等于 19；结点 A 为取极小值的 min 结点，因此 A 的值只能取 B 的值 10，也就是说不再需要求结点 C 的子结点 E 和 F 的值就可以得出结点 A 的值。这样将结点 D 的后继兄弟结点减去称为 β 剪枝。



三. 遗传算法

1. 问题描述

TSP 问题（Traveling Salesman Problem）又译为旅行推销员问题、货郎担问题，是数学领域中著名问题之一。假设有一个旅行商人要拜访 n 个城市，他必须选择

所要走的路径，路径的限制是每个城市只能拜访一次，而且最后要回到原来出发的城市。路径的选择目标是要求得的路径路程为所有路径之中的最小值。

注意：本实验要求使用“遗传算法”来解决该问题。

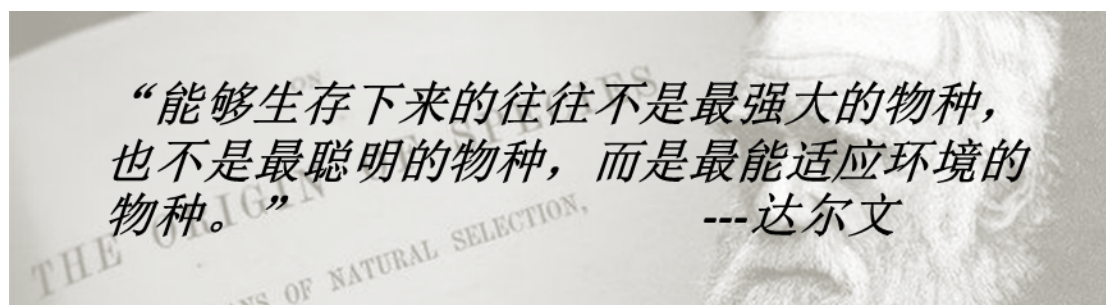
样例

由于本实验采用随机初始化进行输入，所以不设计标准的输入输出样例。

输入：随机初始化 10 个城市的坐标，假设所有城市两两之间均存在交通路线，其长度为两城市之间的欧氏距离。

输出：遍历所有 10 个城市所需要的最短距离，每个城市仅允许拜访一次。

2. 问题分析



遗传算法『适者生存』、『优胜劣汰』的原则，是一类借鉴生物界自然选择和自然遗传机制的随机化搜索算法。遗传算法模拟一个人工种群的进化过程，通过选择(Selection)、交叉(Crossover)以及变异(Mutation)等机制，在每次迭代中都保留一组候选个体，重复此过程，种群经过若干代进化后，理想情况下其适应度达到近似最优的状态。自从遗传算法被提出以来，其得到了广泛的应用，特别是在函数优化、生产调度、模式识别、神经网络、自适应控制等领域，遗传算法发挥了很大的作用，提高了一些问题求解的效率。

第一步，初始化：初始化群体一般都是随机生成一个规模为 N 的初始群体。在这里，可以定义一个 s 行 $(t+1)$ 列的 pop 矩阵来表示群体， t 为城市个数， $+1$ 是为了放适应度得分， s 为样本中个体数目。适应度函数的设计是根据个体适应

值对其优劣判定的评价函数。在该问题中用距离的总和作为适应度函数，来衡量求解结果是否最优。

第二步，选择：以一定的概率从群体中选择优胜个体的操作，它是建立在群体中个体适应度评估基础上的。为了加快局部搜索的速度，在算法中采用最优保存策略的方法，即将群体中适应度最大的个体直接替换适应度最小的个体。它们不进行交叉和变异运算，而是直接复制到下一代，以免交叉和变异运算破坏种群中的优秀个体。

第三步，产生新个体并变异：交叉算子是产生新个体的主要手段。它是指将个体进行两两配对，以交叉概率 P_a 将配对的父代个体的部分结构加以替换重组生成新个体的操作；变异操作是以较小的概率 P_b 对群体中个体编码串上的某位或者某些位作变动，从而生成新的个体。

多次迭代之后即可产生最终结果。

提示：种群越多，最终效果肯定越好，但是同样需要考虑自身电脑性能是否支持大规模的种群训练；智能化算法的缺点在于随机化，这导致性能效果往往受随机变异的制约，每次迭代情况并不一样，并且**最终结果可能达不到最优解**。

四. 实验提交要求

将实验 1-3 问题分析，实验运行情况按照实验报告样例编写并且保存到以“学号+姓名.doc/docx”形式的文件中，例如“202022408001+张三.doc/docx”。于**10 月 13 号**之前**交给袁雪婵同学即可**。实验报告样例如下：

封面（提交时删除）

《人工智能》实验报告

实验名称 知识表征方法（每次实验自行更改）

学 号 _____

姓 名 _____

日期

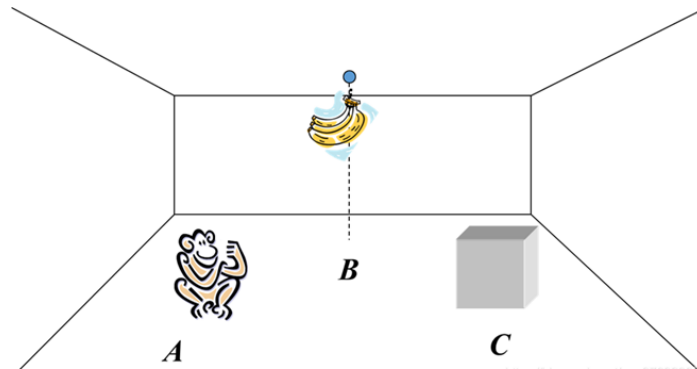
内页：（提交时删除）

实验一 知识表示方法

一、猴子摘香蕉问题

1.实验内容：

房内有一个猴子，一个箱子，天花板上挂了一串香蕉，其位置如图 1 所示，猴子为了拿到香蕉，它必须把箱子搬到香蕉下面，然后再爬到箱子上。请定义必要的谓词，列出问题的初始化状态（即下图所示状态），目标状态（猴子拿到了香蕉，站在箱子上，箱子位于位置 b）。



2.实验思路：

3.程序清单：需加适当注释

4.运行结果说明：

二、.....

...

文字用小 4 号或 4 号；程序和注释用 5 号，程序不能截图，需可再运行。