# 人工智能-第二次课程作业报告

授课教师：张宇 作者：<张博彦>-<58121124>

## 1 问题描述

### 1.1题目介绍



### 1.2任务说明



### 1.3实验环境

Microsoft Visual Studio Community 2022

版本 17.1.2

VisualStudio.17.Release/17.1.2+32319.34

Microsoft .NET Framework

版本 4.8.04084

已安装的版本: Community

Visual C++ 2022 00482-90000-00000-AA689

Microsoft Visual C++ 2022

ASP.NET and Web Tools 2019 17.1.358.51495

ASP.NET and Web Tools 2019

Azure 应用服务工具 3.0.0 版 17.1.358.51495

Azure 应用服务工具 3.0.0 版

C# 工具 4.1.0-5.22165.10+e555772db77ca828b02b4bd547c318387f11d01f

IDE 中使用的 C# 组件。可能使用其他版本的编译器，具体取决于你的项目类型和设置。

Microsoft JVM Debugger 1.0

Provides support for connecting the Visual Studio debugger to JDWP compatible Java Virtual Machines

Microsoft MI-Based Debugger 1.0

Provides support for connecting Visual Studio to MI compatible debuggers

Microsoft Visual C++ 向导 1.0

Microsoft Visual C++ 向导

Microsoft Visual Studio VC 软件包 1.0

Microsoft Visual Studio VC 软件包

NuGet 包管理器 6.1.0

Visual Studio 中的 NuGet 包管理器。有关 NuGet 的详细信息，请访问 https://docs.nuget.org/

TypeScript Tools 17.0.1229.2001

TypeScript Tools for Microsoft Visual Studio

Visual Basic 工具 4.1.0-5.22165.10+e555772db77ca828b02b4bd547c318387f11d01f

IDE 中使用的 Visual Basic 组件。可能使用其他版本的编译器，具体取决于你的项目类型和设置。

Visual Studio Code 调试适配器主机包 1.0

用于在 Visual Studio 中托管 Visual Studio Code 调试适配器的互操作层

Visual Studio IntelliCode 2.2

Visual Studio 的 AI 协助开发。

Visual Studio Tools for CMake 1.0

Visual Studio Tools for CMake

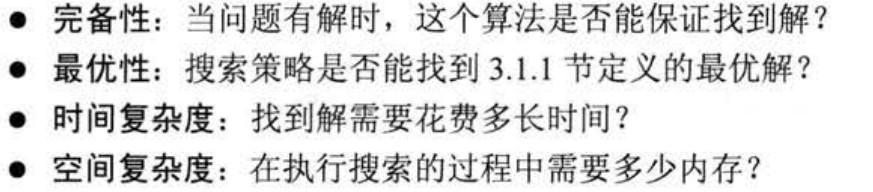
适用于 Google Test 的测试适配器 1.0

启用带有针对 Google Test 编写的单元测试的 Visual Studio 测试工具。扩展安装目录中提供了使用条款和第三方通知。

用于 Boost.Test 的测试适配器 1.0

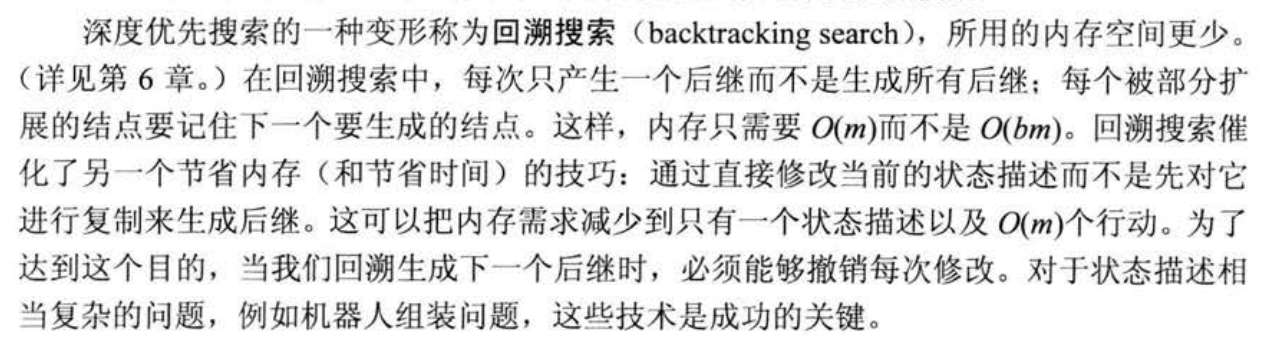
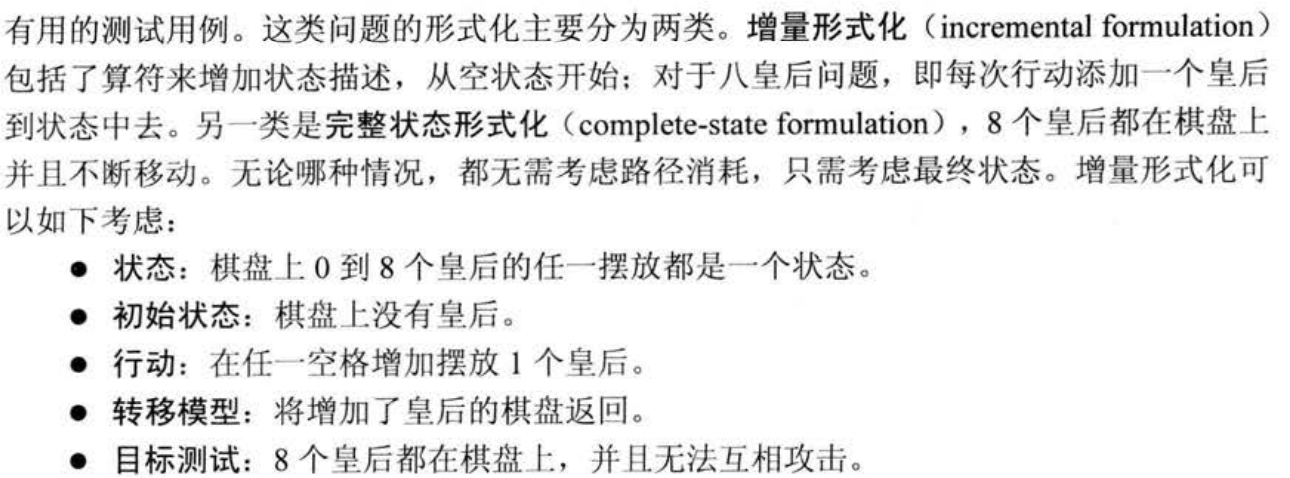
通过针对 Boost.Test 编写的单元测试启用 Visual Studio 测试工具。扩展安装目录中提供用户条款和第三方通知。

### 1.4评价标准

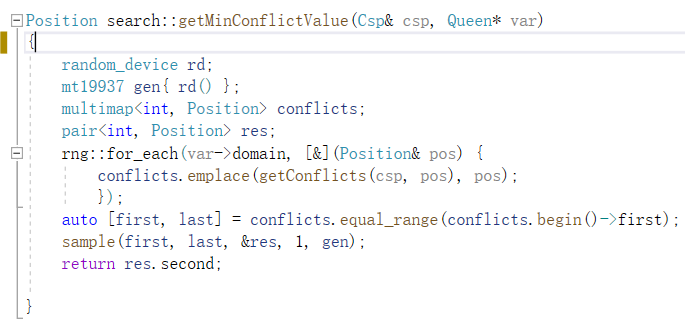
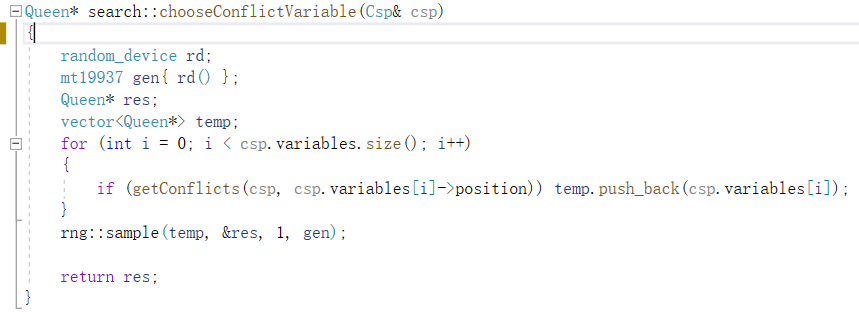
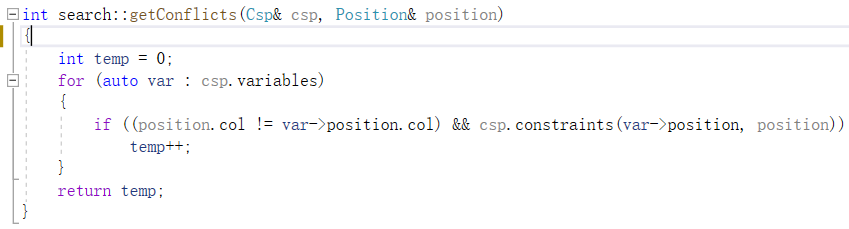
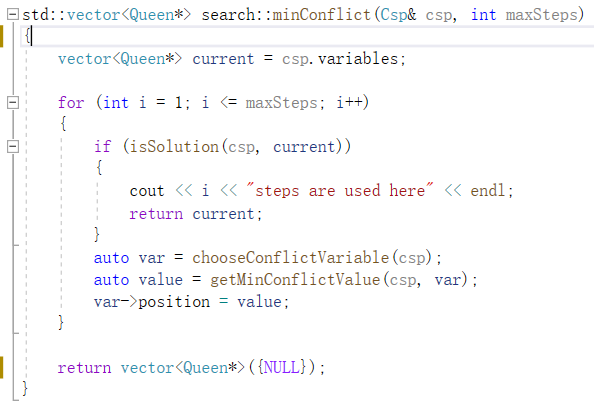


## 2 实验方案

**2.1理论基础**

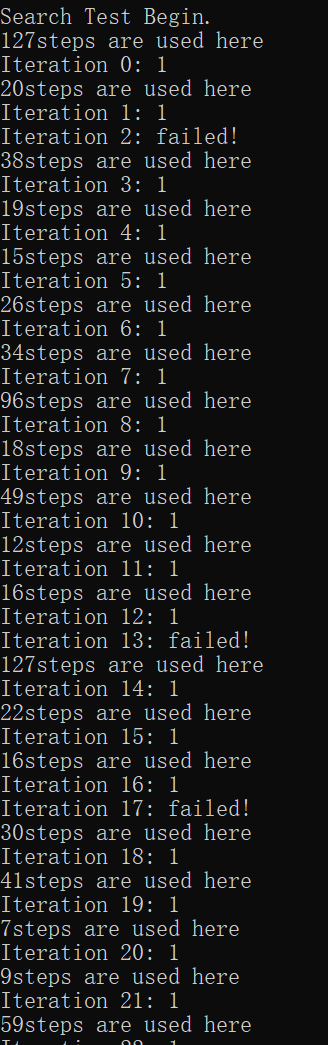
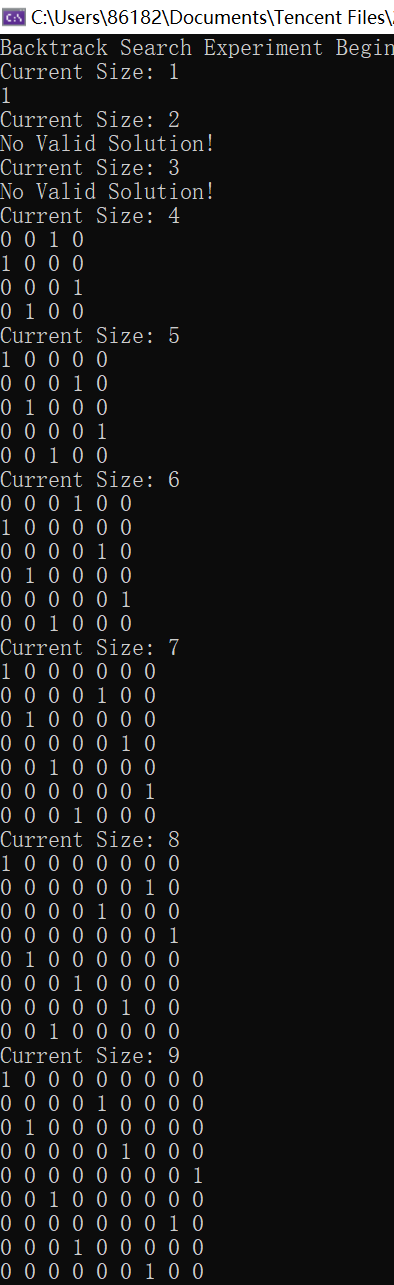


**2.2具体代码**

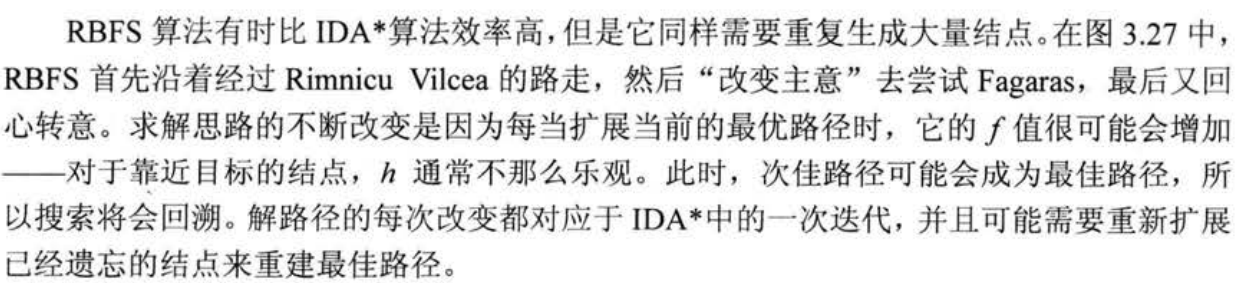
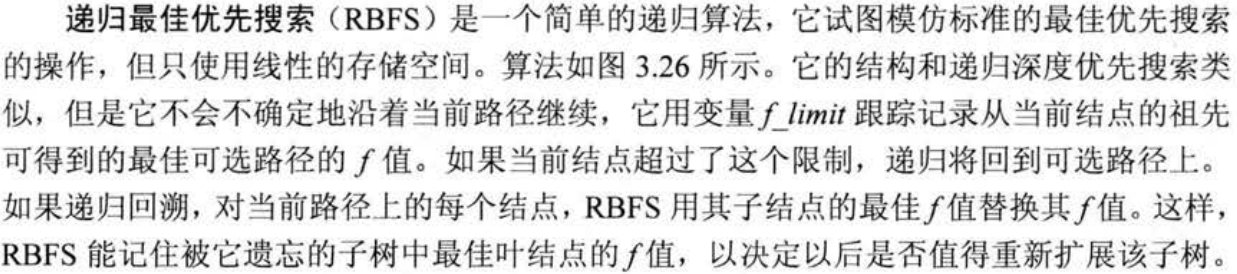


## 3实验结果

部分实验结果展示：



## 4实验分析



## 5 结论

在回溯法中，每次扩大当前部分解时，都面临一个可选的状态集合，新的部分解就通过在该集合中选择构造而成。这样的状态集合，其结构是一棵多叉树，每个树结点代表一个可能的部分解，它的儿子是在它的基础上生成的其他部分解。树根为初始状态，这样的状态集合称为状态空间树。

回溯法对任一解的生成，一般都采用逐步扩大解的方式。每前进一步，都试图在当前部分解的基础上扩大该部分解。它在问题的状态空间树中，从开始结点（根结点）出发，以深度优先搜索整个状态空间。这个开始结点成为活结点，同时也成为当前的扩展结点。在当前扩展结点处，搜索向纵深方向移至一个新结点。这个新结点成为新的活结点，并成为当前扩展结点。如果在当前扩展结点处不能再向纵深方向移动，则当前扩展结点就成为死结点。此时，应往回移动（回溯）至最近的活结点处，并使这个活结点成为当前扩展结点。回溯法以这种工作方式递归地在状态空间中搜索，直到找到所要求的解或解空间中已无活结点时为止。

但当puzzle的规模很大时，使用回溯法的搜索耗时巨大，因此回溯法也需要根据实际情况来确定是否使用。