

Homework #1: State Space Implementation

Submit Assignment

程序代写代做 CS编程辅导

Due Apr 2 by 11:59pm

Points 100

Submitting a file upload

In this assignment you will implement a sliding tile puzzle state space.

This homework is to be completed individually (3703) or in groups of up to two people (3703). You can discuss the assignment with your group, but all programming and other portions must be entirely your own work. You may not show your code to other students. If you are working with another student, you need to designate this by forming a group in Canvas.

The assignment can be implemented in any language, but sample code will be provided in C++. Your assignment should be uploaded to Canvas before the deadline.

Implement your code as follows:

1. Implement a simple data structure to store/represent a single state.
 1. States should either be initialized to the goal state, or should be constructed as a copy of an existing state.
 2. The goal state has the tiles sorted from 0...14, where the 0 tile is the blank.
 3. Implement an equality operator to test if two states are equal.
 4. Store the current location of the blank tile in the state to make state space operations less expensive.
2. Implement a simple data structure (or enum) to represent an operator.
3. Implement a class/methods to operate on states, providing:
 1. a successor function
 2. an operator function
 3. functions for applying and undoing operators

Your environment might look something like this:

```
class STP {
public:
    void GetSuccessors(STPState &nodeID, std::vector<STPState> &actions);
    void GetOperators(STPState &nodeID, std::vector<STPSlideDir> &operators);
    void ApplyOperator(STPState &s, STPSlideDir o);
    void UndoOperator(STPState &s, STPSlideDir o);
};
```

Once your code is implemented, write two versions of the function DoRandomWalk, which takes as input a STP class, a STPState, and a walk distance (n). The function should reset to the goal state, perform n random actions, and then return the resulting state.

The first version should use the GetSuccessors call, while the second version should use a sufficiently large random walk (e.g. $n = 100000000$), time the difference between the two, and report which one is faster.

Include a README.TXT file with your timing results in your solution.

Course Chat

2

people
online

New
message
alerts



Homework 1: State Space Implementation



STPState class implementation

States should either be initialized or should be constructed as a copy of an existing state. The tiles are sorted from 0...14, 0 tile is the blank. Implement an equality operator to test if two states are equal. Store the current location of the blank tile in the state to make state space operations less expensive.

Operator data structure (or enum) implementation

Class or set of methods to operate on STPStates

A successor function, operator function, and functions for applying and undoing operators

DoRandomWalk function version 1 implementation

DoRandomWalk version 1 takes as input a STP class, a STPState, and a walk distance (n). The function should reset to the goal state, perform n random actions, and then return the resulting state. Version 1 should use the GetSuccessors call.

DoRandomWalk function version 2 implementation

DoRandomWalk version 2 takes as input a STP class, a STPState, and a walk distance (n). The function should reset to the goal state, perform n random actions, and then return the resulting state. Version 2 should use the GetActions (GetOperators) call.

Time the difference between Random Walk version 1 and version 2 and report which one is faster.

README.txt included with timing results

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

https://tutorcs.com

Marks	Marks	
15.0 pts Full Marks	0.0 pts No Marks	15.0 pts
15.0 pts Full Marks	0.0 pts No Marks	15.0 pts
5.0 pts Full Marks	0.0 pts No Marks	5.0 pts
5.0 pts Full Marks	0.0 pts No Marks	5.0 pts

Total Points: 100.0