CS-461

Artificial Intelligence

Homework #3

Group Name: Enigma
Group Members:
1. Ege Yosunkaya - 21402025 / CS
2. Kerem Ayöz - 21501569 / CS
3. Erkut Alakuş - 21101413 / CS

Code

BBS

```
class BBS:

    # Initializer
    def __init__(self, initialState):
        self.initialState = initialState
        self.goalState = [1, 2, 3, 8, -1, 4, 7, 6, 5]
        path = [initialState]
        self.queue = [path]
        self.closedList = []

    # Sort the paths by their total length + distance to the goal
    def sortQueue(self):
        for passnum in range(len(self.queue) - 1, 0, -1):
            for i in range(passnum):
                if len(self.queue[i]) > len(self.queue[i + 1]):
                    temp = self.queue[i]
                    self.queue[i] = self.queue[i + 1]
                    self.queue[i + 1] = temp

    def checkGoal(self):
        if not self.queue:
            print("Success \n")
            print(self.queue[0])

    def contains(self, state):
        for i in range(len(self.closedList)):
            if self.closedList[i].compare(state):
                return True
        return False

    def removeLoops(self, states):
        newStates = copy.deepcopy(states)
        length = len(states)
        for i in range(0, length):
            for closedState in self.closedList:
                if newStates[i].compare(closedState):
                    del newStates[i]
                    length -= 1

        return newStates
```

```
def solve(self):
    while self.queue:
        if self.queue[0][-1].distanceToGoal() == 0:
            return self.queue[0]
        # Extend the front path's last state
        newStates = self.queue[0][-1].makeAllMoves()
        for i in range(0, len(self.queue[0]) - 1):
            for newState in newStates:
                if self.queue[0][i].compare(newState):
                    newStates.remove(newState)
        # For every possible extension
        for state in newStates:
            # Append the new path
            temp = copy.deepcopy(self.queue[0])
            temp.append(state)
            self.queue.append(temp)

        # Delete the extended path
        del self.queue[0]

        # Sort the queue
        self.sortQueue()
        # print(self.queue[0][-1])
```

BBSDP

```
class BBSDP(BBS):
    def solve(self):
        while self.queue:
            if self.queue[0][-1].distanceToGoal() == 0:
                return self.queue[0]

            # Extend the front path's last state
            newStates = self.queue[0][-1].makeAllMoves()

            for i in range(0, len(self.queue[0]) - 1):
                for newState in newStates:
                    if self.queue[0][i].compare(newState):
                        newStates.remove(newState)
            # For every possible extension
            for state in newStates:
                # If the extended state is unique
                if not self.contains(state):
                    # Append new path to the queue
                    temp = copy.deepcopy(self.queue[0])
                    temp.append(state)
                    self.queue.append(temp)
                    self.closedList.append(state)

                # Else, compare it with the same states in the queue(not necessarily the end
nodes in the paths in queue)
                else:
                    # Cost of the path that is going to be added
                    l1 = len(self.queue[0]) + 1
                    # Check the queue if it contains the same state with the path's last state
                    for path2 in self.queue:
                        # Try to find the common node in every path in queue, it may not be
necessarily the last state of the current paths
                        for i in range(len(path2), 0):
                            # If found
                            if path2[i].compare(state) and l1 < i + 1:
                                # Create the newly added path
                                temp = copy.deepcopy(self.queue[0])
                                temp.append(state)
                                # Remove the most cost path
                                self.queue.remove(path2)
                                # Append the new least cost path
                                self.queue.append(temp)
                                break
            del self.queue[0]
            self.sortQueue()
```

For same initial state BBS and BBSDP solutions

BBS Solution

```
--------------
| 8 | 1 | -1 |
-------------
| 2 | 4 | 3 |
-------------
| 7 | 6 | 5 |
-------------
Distance to goal: 12

--------------
| 8 | 1 | 3 |
-------------
| 2 | 4 | -1 |
-------------
| 7 | 6 | 5 |
-------------
Distance to goal: 8

--------------
| 8 | 1 | 3 |
-------------
| 2 | -1 | 4 |
-------------
| 7 | 6 | 5 |
-------------
Distance to goal: 6

--------------
| 8 | 1 | 3 |
-------------
| -1 | 2 | 4 |
-------------
| 7 | 6 | 5 |
-------------
Distance to goal: 8

--------------
| -1 | 1 | 3 |
-------------
| 8 | 2 | 4 |
-------------
| 7 | 6 | 5 |
-------------
Distance to goal: 8

--------------
| 1 | -1 | 3 |
-------------
| 8 | 2 | 4 |
-------------
| 7 | 6 | 5 |
-------------
Distance to goal: 6


--------------
| 1 | 2 | 3 |
-------------
| 8 | -1 | 4 |
-------------
| 7 | 6 | 5 |
-------------
Distance to goal: 0
```

BBSDP Solution

```
--------------
| 8 | 1 | -1 |
-------------
| 2 | 4 | 3 |
-------------
| 7 | 6 | 5 |
-------------
Distance to goal: 12
--------------
| 8 | 1 | 3 |
-------------
| 2 | 4 | -1 |
-------------
| 7 | 6 | 5 |
-------------
Distance to goal: 8
--------------
| 8 | 1 | 3 |
-------------
| 2 | -1 | 4 |
-------------
| 7 | 6 | 5 |
-------------
Distance to goal: 6
--------------
| 8 | 1 | 3 |
-------------
| -1 | 2 | 4 |
-------------
| 7 | 6 | 5 |
-------------
Distance to goal: 8
--------------
| -1 | 1 | 3 |
-------------
| 8 | 2 | 4 |
-------------
| 7 | 6 | 5 |
-------------
Distance to goal: 8
--------------
| 1 | -1 | 3 |
-------------
| 8 | 2 | 4 |
-------------
| 7 | 6 | 5 |
-------------
Distance to goal: 6
--------------
| 1 | 2 | 3 |
-------------
| 8 | -1 | 4 |
-------------
| 7 | 6 | 5 |
-------------
Distance to goal: 0
```

<u>BBS and BBSDP move counts for 25 initial states</u>

```
--------------
| 1 | 2 | 3 |
-------------
| -1 | 6 | 4 |
-------------
| 8 | 7 | 5 |
-------------
BBS move count: 4
BBSDP move count: 4

--------------
| 1 | 2 | 3 |
-------------
| 8 | 6 | 4 |
-------------
| 7 | -1 | 5 |
-------------
BBS move count: 2
BBSDP move count: 2

--------------
| 1 | 3 | 4 |
-------------
| 8 | 2 | 5 |
-------------
| 7 | 6 | -1 |
-------------
BBS move count: 5
BBSDP move count: 5

--------------
| 2 | 6 | 3 |
-------------
| 1 | 7 | -1 |
-------------
| 8 | 5 | 4 |
-------------
BBS move count: 10
BBSDP move count: 10
```

```
--------------
| 2 | 8 | 3 |
-------------
| 1 | -1 | 4 |
-------------
| 7 | 6 | 5 |
-------------
BBS move count: 5
BBSDP move count: 5

--------------
| 8 | 1 | 3 |
-------------
| -1 | 2 | 4 |
-------------
| 7 | 6 | 5 |
-------------
BBS move count: 4
BBSDP move count: 4

--------------
| 1 | 2 | 3 |
-------------
| 7 | 8 | 6 |
-------------
| -1 | 5 | 4 |
-------------
BBS move count: 7
BBSDP move count: 7


--------------
| 2 | 8 | 3 |
-------------
| -1 | 1 | 4 |
-------------
| 7 | 6 | 5 |
-------------
BBS move count: 6
BBSDP move count: 6
```

```
--------------
| 1 | 4 | 2 |
-------------
| 8 | 3 | 5 |
-------------
| 7 | 6 | -1 |
-------------
BBS move count: 7
BBSDP move count: 7

--------------
| 1 | 2 | 3 |
-------------
| 8 | 4 | 5 |
-------------
| 7 | 6 | -1 |
-------------
BBS move count: 3
BBSDP move count: 3

--------------
| 1 | 2 | 3 |
-------------
| -1 | 8 | 4 |
-------------
| 7 | 6 | 5 |
-------------
BBS move count: 2
BBSDP move count: 2

--------------
| 1 | 2 | 3 |
-------------
| 6 | 7 | 8 |
-------------
| 5 | -1 | 4 |
-------------
BBS move count: 10
BBSDP move count: 10
```

```
--------------
| 1 | 2 | 3 |
-------------
| 8 | 6 | 4 |
-------------
| -1 | 7 | 5 |
-------------
BBS move count: 3
BBSDP move count: 3


--------------
| 8 | 1 | 3 |
-------------
| 7 | 2 | 4 |
-------------
| -1 | 6 | 5 |
-------------
BBS move count: 5
BBSDP move count: 5

--------------
| 1 | 2 | 3 |
-------------
| -1 | 8 | 4 |
-------------
| 7 | 6 | 5 |
-------------
BBS move count: 2
BBSDP move count: 2

--------------
| 1 | 2 | -1 |
-------------
| 8 | 4 | 3 |
-------------
| 7 | 6 | 5 |
-------------
BBS move count: 3
BBSDP move count: 3
```

```
--------------
| 2 | 3 | 4 |
-------------
| -1 | 8 | 5 |
-------------
| 1 | 7 | 6 |
-------------
BBS move count: 10
BBSDP move count: 10

--------------
| 2 | 8 | -1 |
-------------
| 1 | 4 | 3 |
-------------
| 7 | 6 | 5 |
-------------
BBS move count: 7
BBSDP move count: 7

--------------
| 2 | -1 | 3 |
-------------
| 1 | 6 | 4 |
-------------
| 8 | 7 | 5 |
-------------
BBS move count: 6
BBSDP move count: 6

--------------
| 1 | 4 | 2 |
-------------
| 8 | 3 | -1 |
-------------
| 7 | 6 | 5 |
-------------
BBS move count: 6
BBSDP move count: 6

--------------
| 2 | 3 | -1 |
-------------
| 1 | 8 | 4 |
-------------
| 7 | 6 | 5 |
-------------
BBS move count: 5
BBSDP move count: 5

--------------
| 1 | 2 | 3 |
-------------
| 7 | 8 | 4 |
-------------
| -1 | 6 | 5 |
-------------
BBS move count: 3
BBSDP move count: 3

--------------
| 1 | 2 | 3 |
-------------
| 6 | 7 | 4 |
-------------
| -1 | 8 | 5 |
-------------
BBS move count: 7
BBSDP move count: 7

--------------
| 1 | 4 | 2 |
-------------
| 8 | 6 | 3 |
-------------
| 7 | -1 | 5 |
-------------
BBS move count: 6
BBSDP move count: 6

--------------
| 1 | 2 | 3 |
-------------
| 8 | 6 | 4 |
-------------
| 7 | 5 | -1 |
-------------
BBS move count: 3
BBSDP move count: 3
```

Figures

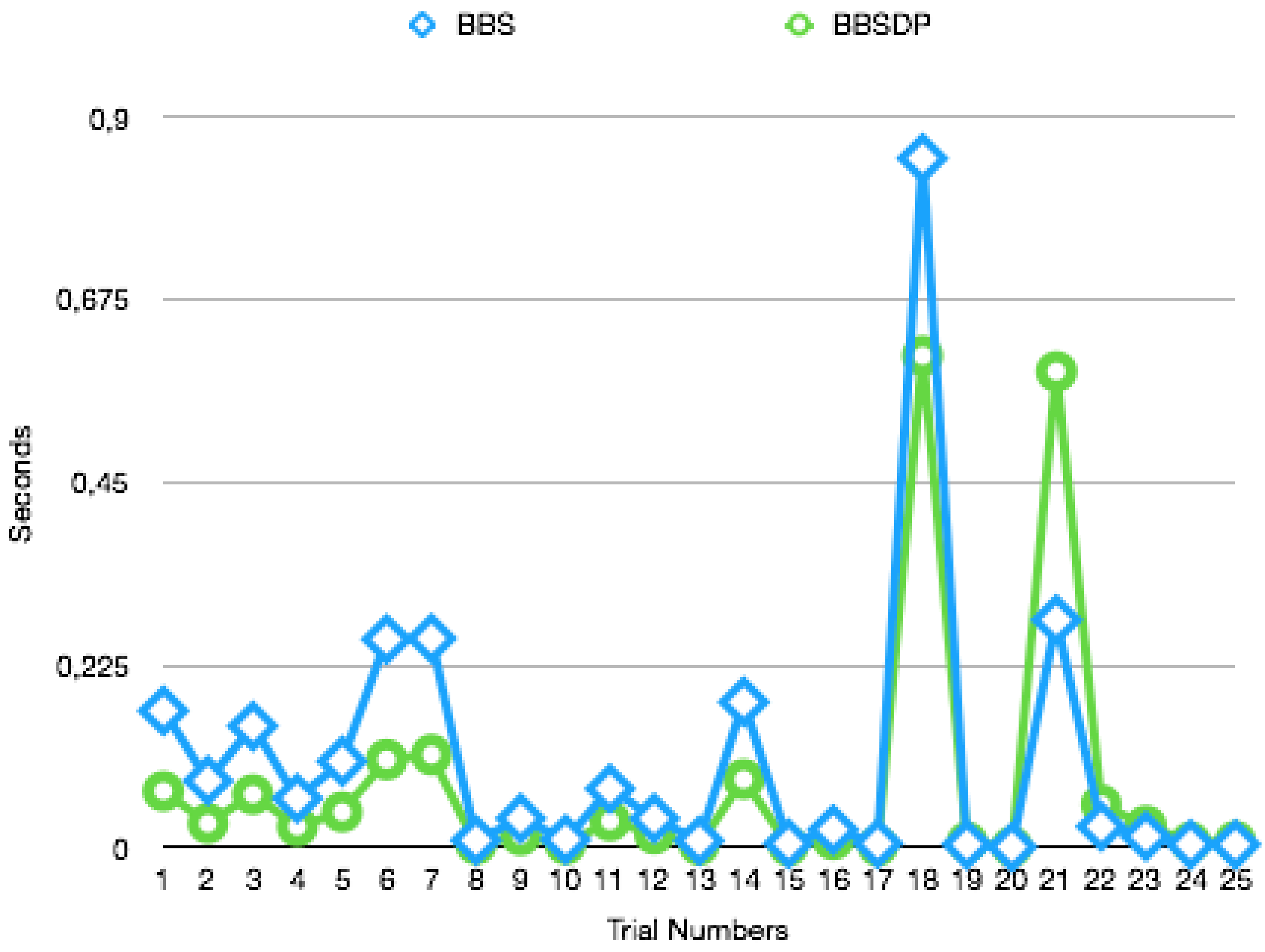

Figure1: CPU time(seconds) vs Trial Number

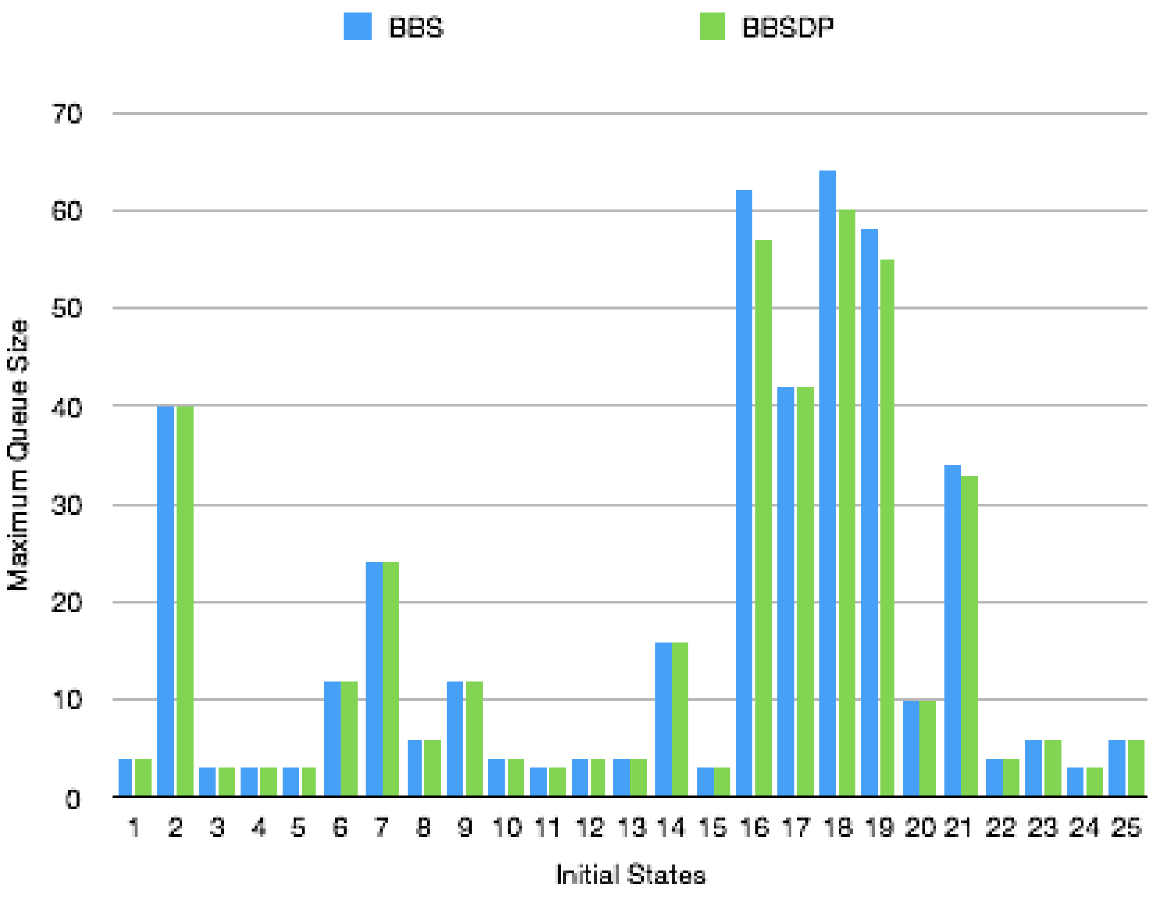

Figure 2: Maximum Queue Size vs States Plot