

Homework #1: Search Algorithms

Assigned: 19.11.2020 Due: 10.12.2020

1. Objective

The purpose of this homework is to teach you how to apply uninformed and informed search algorithms while solving a real problem and how to evaluate the performance of these algorithms.

2. Specification

In this homework, you will formulate and develop a solution to the below problem by applying uninformed and informed search algorithms discussed in class: Breadth First Search, Depth First Search, Depth Limited Search, Iterative Deepening Search, Uniform Cost Search, Greedy Best First Search, and A* Search.

Pancake problem¹

This famous problem is usually stated as follows: Pancake sorting is sorting a disordered stack of pancakes in order of size when a spatula can be inserted at any point in the stack and used to flip all pancakes above it. (See Figure 1 below and L4_InformedSearching slides.)

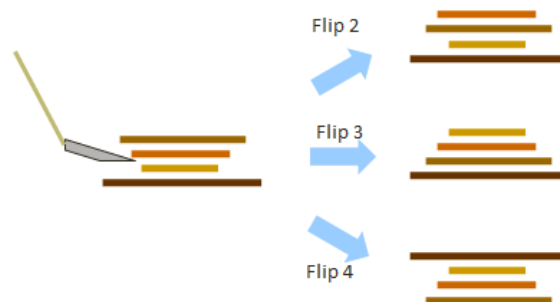


Figure 1 – Pancake sorting problem illustrated.

Task 1 – Implementation

In this homework, you will use **simpleai** library. You are not allowed to use any other AI library and you are not allowed to implement your own search algorithms or pancake sorting. Your job is to formulate and develop a solution to the pancake problem as a search problem.

1. Your program should have a simple console-based menu which takes the *number of pancakes* (N) from the user. Assume that the smallest pancake has number 0 and the largest one has $N-1$.
2. You should also ask two options to the user for the initial ordering of N pancakes:

¹ https://en.wikipedia.org/wiki/Pancake_sorting

- a. The program randomly generates a permutation of N pancakes as the initial ordering. (A sample run is shown below)

```
Enter number of pancakes: 6
Do you want to enter ordering?: no
initial state: (0, 3, 4, 5, 2, 1)
```

- b. The initial ordering of N pancakes is determined by the user. (A sample run is shown below)

```
Enter number of pancakes: 5
Do you want to enter ordering?: yes
Enter top to bottom ordering between [0 - 4], separated by spaces: 3 1 0 4 2
initial state: (3, 1, 0, 4, 2)
```

3. Then using simpleai library (you will need *traditional.py* and *models.py*), define the problem as a search problem (implement required functions such as actions, result, is_goal, etc.) and call 7 search algorithms listed above to solve the problem.
4. You can get help from sample codes (hello_world, eight_puzzle, missionaries, game_walk) of simpleai library to learn how to define a problem as a search problem.

Task 2 – Report

You will also write a report including the following sections:

A – Problem formulation

In this part, clearly explain the problem formulation including the *initial state*, *possible actions*, *transition model*, *goal test*, and *path cost*.

B – Heuristic functions

In informed search algorithms, you will need a heuristic function. In this part, describe at least one heuristic function for the given problem. Try to discuss the admissibility and consistency properties of your function. (Give proper citation if you take anything from Internet.)

C – Discussion on the results

This part is the most important part of the report and homework. Run your implementation with different parameters and inputs (number of pancakes, initial ordering, depth limit for DLS, etc.). Discuss the results of different search algorithms in terms of *completeness*, *optimality*, and *time and space complexity* by giving code outputs. If you developed more than one heuristic function, compare them too. Compare the uninformed search algorithms to informed search algorithms. Use graph search mainly, but you can also compare graph search and tree search versions of the algorithms. You can try following cases as initial pancake ordering (top to bottom order):

- 5 2 3 0 1 4
- 6 1 0 5 2 4 3
- 7 6 4 5 0 2 1 3

3. Submission

- This homework can be done in groups of maximum 3.
- You will submit your report in pdf format and source files (only the code you implemented, not simpleai library codes).
- Place all your files in a zip archive with name **HW1_StudentID1_ StudentID2_ StudentID3.zip** and submit through the MS Teams submission module.
- Single submission from each group is sufficient.
- If you have further questions, you can send me an e-mail.

4. Academic Honesty Policy

You cannot borrow other's ideas or portions of codes, without giving proper citation. This can be an Internet source or your friend. Clearly indicate which part of your code/report/idea is borrowed from where. Of course, you cannot get all or most of your work from others'. Otherwise you will be penalized.

5. Late Submission Policy

Deadline for homework submissions is **23:59 pm** at the specified date. For each additional day, **25% cut-off** will be applied.

Dr. Zeynep ÇİPİLOĞLU YILDIZ