HW 1-G - Burnt Pancake

Submit Assignment

Due Mar 13 by 11:59pm **Points** 100 **Submitting** a file upload

Intro to Dear AI - Spring 2020

Homework 1 – Grad version

Burnt pancake problem

Due: Friday, March 13th, 11:59 pm

In Short

Write a program that receives an order of 4 bottom-burnt pancakes and prints the solution that BFS and A* search will find for going from the Start state to the Goal (ordered pancakes and all burnt-side down).

Description

In this programming homework, you will work on a modified version of the classic pancake problem (https://en.wikipedia.org/wiki/Pancake_sorting#The_original_pancake_problem) that was discussed in the class. In this version, one side of each pancake is burnt, and the pancakes must be sorted with all the burnt-side down. You need to write a program that receives an arbitrary order of such 4 pancakes from the user, plus the type of the search algorithm, and prints the steps that the specified algorithm will take to reach the Goal state.

Format

Similar to the format described in class, each of the pancakes has an ID number that is consistent with their size followed by a letter “w” or “b”. This way, the largest pancake has an ID of 4, the next largest 3, next 2, and the smallest has an ID of 1. The letter “w” refers to the unburnt (white) side is up, and “b” shows that the burnt side is up. The goal is reaching to “1w2w3w4w”. For instance, Fig. 1 shows an example of the IDs associated with each pancake in a certain configuration.

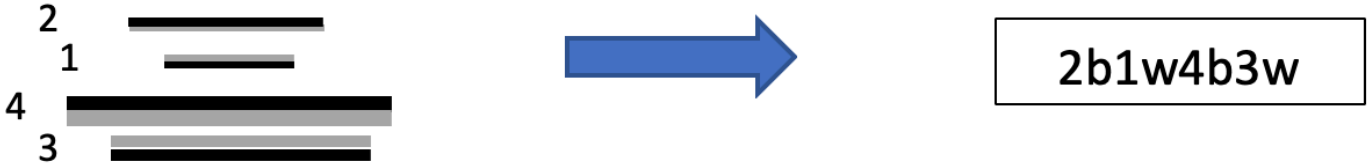

Figure 1 – Pancakes’ representation. Grey color refers to unburnt (white) part and black refers to the burnt part.

Input

The input should consist of pairs of four digits and one character, a hyphen, and one last character (#C#C#C#C-X), where the first digit indicates the ID of the top pancake and the first character indicates whether if the burnt side is down (“w”) or not (“b”), the second number indicates the second-highest pancake followed by a character, etc. The last character (X) would be either one of “b” or “a” characters, which refer to the Breadth-First (BFS) and A* search algorithms respectively.

Implementation

The cost associated with each flip is equal to the number of pancakes that are being flipped. For instance, the cost of one flip between pancake 3b and 2b from the state “4w1b2w3b” to “2b1w4b3b” is equal to 3 (spatula between 2 and 3). For each state, use the same heuristic function (h(x)) that was discussed in the class:

“the ID of the largest pancake that is still out of place”. For BFS, you don’t need to consider a cost and a heuristic function. Use the graph version of the algorithms, meaning that use some type of list (closed set) to avoid visiting the nodes multiple times.

Add as many comments as you can to your code, so that it’s easy to understand your implementation, including the role of functions, variables, etc. Specifically, make it clear how your fringe is implemented and employed. Use an informative name for your fringe and add comments where you define that.

Tie-Breaking

When needed for any of the search algorithms, use the following tie-breaking mechanism:

"when there is a tie between two nodes, replace “w” with 1 and “b” with 0 to obtain an eight-digit number. After that pick the node with a larger numerical ID chosen."

For instance, if there is a tie between 4b3w2b1b and 3w4w2b1b, then 4b3w2b1b will be chosen as 40312010>31412010.

Output

Your program must print the steps that the specified algorithm (e.g., BFS) finds to solve the problem. In other words, it simply prints the solution that the algorithm finds. For each state (except the final state), use the character “|” to show where the flip to go to the next step happens. For A*, also print the value for the actual cost (function g), and the value of the heuristic function (function h) in each step. The following is an example of an input and output of the program.

Input:

1b2b3b4w-a # “a” indicates A*

Output (possible):

1b|2b3b4w g=0, h=23 # put the spatula between 1 and 2 to go to next

1w2b|3b4w g=1, h=22

2w1b3b|4w g=3, h=21

3w1w|2b4w g=6, h=19

1b3b2b|4w g=8, h=16

2w3w|1w4w g=11, h=14

3b2b1w|4w g=13, h=12

1b|2w3w4w g=16, h=9

1w2w3w4w g=17, h=0

Note that the values for g and h correspond to the “*current”* state, and the character “|” denotes the location of the flip for going to the “*next”* state. In the above example, “g” and “h” values are for illustration purposes and might be not the correct values.

Programming Language

As previously discussed you can use Python and any language that the TA (if needed, ask on Piazza).

Checking Results

You are allowed and encouraged to post your results (input-output pairs) on Piazza to double-check them with other students.

User Interface (optional)

You can also use a visual user interface (like textbox and buttons) for input and output. Make sure it's clear how the program operates and how the output can be interpreted. The steps taken to reach the goal have to be clear from the output.

Grading

The total points will be 100. For each of the two requested search algorithms, you will receive 50 points. 40 points will be for the code, and 10 for the correct output.

Use a different function for each of the search algorithms, even if it needs copying the same piece of code. Use easily distinguishable names for your functions.

Questions?

Post them on **Pizza** **(https://piazza.com/class/k6dtwl9cuzo2wi)** .

Final Note

Don't start late! It might take you longer than you expect to finish the homework.