

CS C 421 Artificial Intelligence: Fall 2019

Assignment 1

Due: Oct 3 (11:55 pm), (e-submission only, submit code and printout of results)

1. (2 pts) Complete the accompanying base code. It has implementations for several search strategies, but not all, e.g. iterative deepening is not implemented (you should implement it). Look for “TODO” comments.

Consider the following problems. Model them as search problems, implement them using the accompanying code and find a solution using all the search algorithms that work on the particular problem. Report solution cost and number of expansions for each search algorithm that is able to (quickly) terminate on a problem. For each problem (except Water Jugs Problem), find a heuristic function that improves the performance of A* compared to UCS. See the implemented NPuzzle problem in the accompanying code for an example.

2. (5 pts) **Wolf-goat-cabbage problem**

You are on the bank of a river with a boat, a cabbage, a goat, and a wolf. Your task is to get everything to the other side.

Restrictions:

1. only you can handle the boat
2. when you're in the boat, there is only space for one more item
3. you can't leave the goat alone with the wolf, nor with the cabbage (or something will be eaten).

3. (5 pts) **Missionaries and cannibals problem**

Three missionaries and three cannibals seek to cross a river, say from the left bank to the right bank. A boat that may be navigated by any combination of one or two people is available on their side of the river. If cannibals outnumber the missionaries on either side of the river at any time, the cannibals will indulge in their anthropophagic tendencies and do away with the missionaries. Find a sequence of boat trips that will permit all the missionaries and cannibals to cross the river safely.

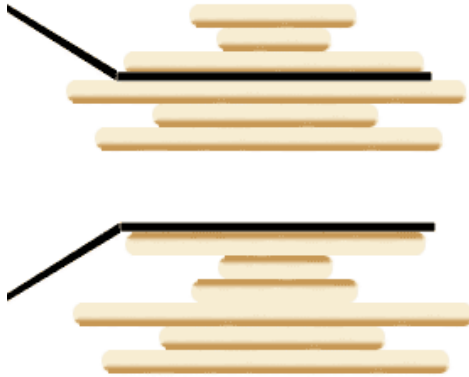
4. (5 pts) **Water Jugs Problem**

You have three jugs, measuring 12 gallons, 8 gallons, and 3 gallons, and a water faucet. You can fill the jugs up, or empty them out from one another or onto the ground. *You need to measure out exactly one gallon.*

Consider a variant of the Water Jugs Problem in which we are interested in minimizing the amount of water that must be poured. Filling the large jug when it is empty requires pouring 12 gallons into it. Likewise, transferring water from a full large jug into an empty small jug requires pouring 3 gallons. A lazy person might not mind a longer solution to the problem if it in fact reduced the amount of water they had to heft and pour.

5. (5 pts) Pancake Sorting Problem

Given a stack of pancakes of various sizes, can you sort them into a stack of decreasing sizes, largest on bottom to smallest on top? You have a spatula with which you can flip the top i pancakes. This is shown below for $i = 3$; on the top the spatula grabs the first three pancakes; on the bottom we see them flipped:



How many flips will it take to get the whole stack sorted? This is an interesting [problem](#) that Bill Gates has [written about](#). A reasonable heuristic for this problem is the *gap heuristic*: if we look at neighboring pancakes, if, say, the 2nd smallest is next to the 3rd smallest, that's good; they should stay next to each other. But if the 2nd smallest is next to the 4th smallest, that's bad: we will require at least one move to separate them and insert the 3rd smallest between them. The gap heuristic counts the number of neighbors that have a gap like this. In our specification of the problem, pancakes are ranked by size: the smallest is 0, the 2nd smallest 1, and so on, and the representation of a state is a tuple of these rankings, from the top to the bottom pancake. Thus the goal state is always $(0, 1, \dots, n-1)$ and the initial (top) state in the diagram above is $(1, 0, 3, 5, 2, 4)$.