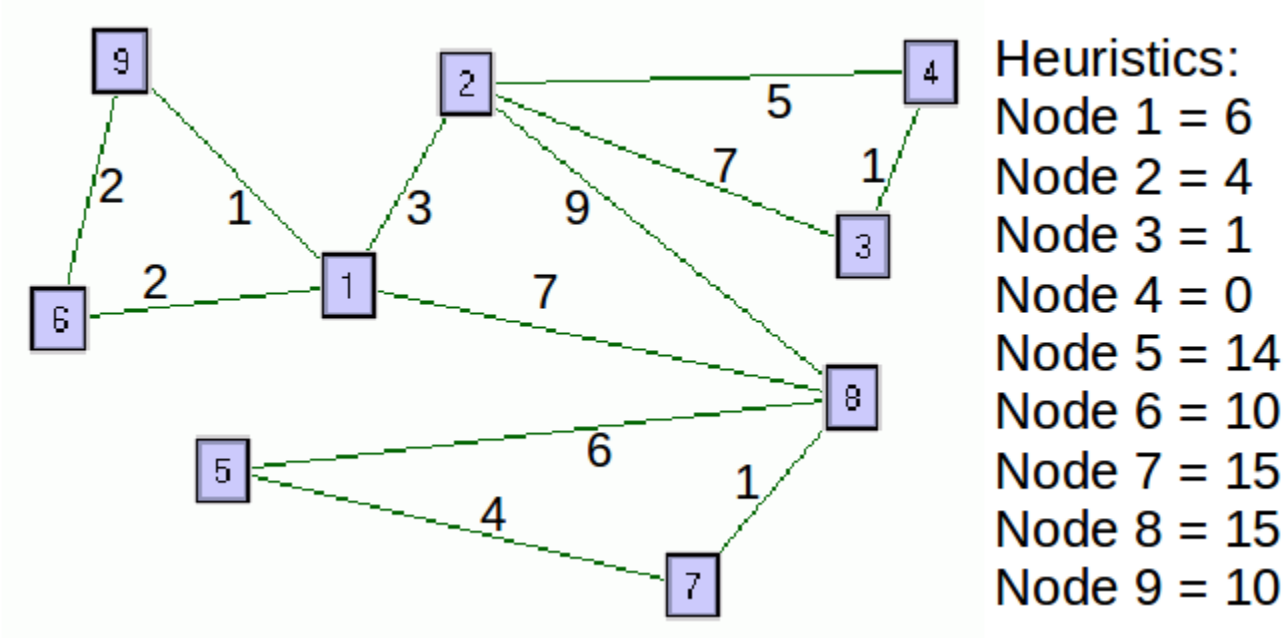
4511W, Spring-2018
ASSIGNMENT 1 :
Assigned: 02/11/18 Due: 02/18/18 at 11:55 PM (submit via moodle, you may scan or take a picture of your paper answers in a zip if you have multiple files)

Problem 1. (15 points)
Perform A* search on the following graph. At each step show (1) the fringe with the corresponding f-values and (2) the explored set. The initial state is vertex "5" and the goal is vertex "4".



Problem 2. (20 points)
(1) Is the heuristic in problem 1 admissible? Is it consistent?

(2) Suppose you had two heuristics. The first heuristic, h_1, is admissible and consistent. The second, h_2, is admissible but **not** consistent. Prove or disprove whether a new heuristic defined as: $h_3 = \min(h_1,h_2)$, is admissible and/or consistent.

Problem 3. (30 points)
For each of the following problems, give: (1) the relaxation, (2) a general description of how to solve it optimally and (3) how to compute the heuristic for a given state (i.e. how to get a value from a state that is not computationally intensive.)

(1) Suppose you have three kids that need to get to school and your car is broken so you have to bike. Unfortunately, you can only carry one child on the bike safely so you will have to make multiple trips. Your children are also not the best behaved. When you are not watching the children, if the oldest and middle child are together they fight. Also if you are not watching, the middle child will sit on the youngest child. The goal is to get all three children to school without them fighting/sitting on each other.

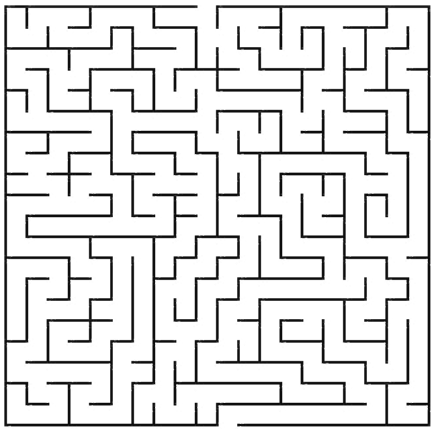
(2) You are playing puzzle-loop: https://www.puzzle-loop.com/faq.php . Suppose you know some vertex that is guaranteed to have an edge leaving it.

Problem 4. (20 points)
For each of the problems, (1) state what algorithm you think is best search from uniformed, informed and local searches. Provide a specific search algorithm and not just one of these categories. Then (2) justify your answer with a **short** paragraph (one to three sentences) why you think this is the most appropriate choice.

(1) Suppose a company hired you to make a better worms AI: https://www.youtube.com/watch?v=aczSWJ8-PK0 (you don't need to watch the whole thing, but just to get an idea of the game). The goal is to kill all enemy worms before you lose all of yours. For simplicity sake, assume you are playing 2-player only and you have some supporting code already worked out for a bot that can figure out how to position yourself. All you need to do now is figure out which weapon to use on which character. How would you find an efficient way to win against the current AI on one specific level?

(2) An airplane carrying plutonium crashed into the desert. You landed nearby with a bio-hazard suit and a Geiger counter. You need to find and retrieve the plutonium. (You can assume all the plutonium is together at the same spot.)

(3) Suppose you want to find the exit to a simple maze, such as shown below. However, the environment is only partially observable, so you can't see what are in the squares until your algorithm "explores" them. Your goal is to find the shortest path to the exit.



(4) You are going on the road trip and are doing some very meticulous planning. You already know the route you are taking, but you want to optimize fuel costs. You have recorded where all the gas stations are located and the price at each one. You also know on a full tank, which other gas stations you can reach (assume gas use is not variable like in real life).

Programming (python/lisp):

The book provides code for the algorithms presented. For this class, we will use the python version of the code. Download the python code here:
https://github.com/aimacode/aima-python

The code requires python3 to run (but some of the tests are written for python 2). For this assignment, you will need to know how to use the implemented genetic algorithm. Of note are:
/root/search.py
/root/tests/test_search.py

I was having trouble running test_search.py directly, but it will be useful to reference.

Problem 5. (20 points)
For this problem, compare the actual run-time of depth-first search, iterative-deepening depth-first search and genetic algorithms. Do this for the n-queens problem (this is already built into the code).

To get the run-time of code on a cse-labs machine put “time” before the program, so for example:

```
time python3 myFile.py
```

... then look for this in the output:

```
8.736u 0.000s 0:11.75 91.3%	0+0k 0+0io 0pf+0w
```

This corresponds to an 8.736 second run-time(of computation).

Note: For depth-first search, use the “depth_first_tree_search” function, not one of the other variants.

Note2: We mentioned in class that there are many variations of on how you can implement genetic algorithms. Please make sure that you are using the book's version correctly.

Answer the following:
(1) For n=11 run all three tests and report the runtime.

(2) For n=20 find the runtime for depth-first search and genetic algorithms (ID-DFS takes a long time...)

(3) For n=40 find the runtime for the genetic algorithm. (Both DFS and ID-DFS take a long time.)

(4) Write a short paragraph of analysis weighing the pros/cons of the algorithms in this setting.