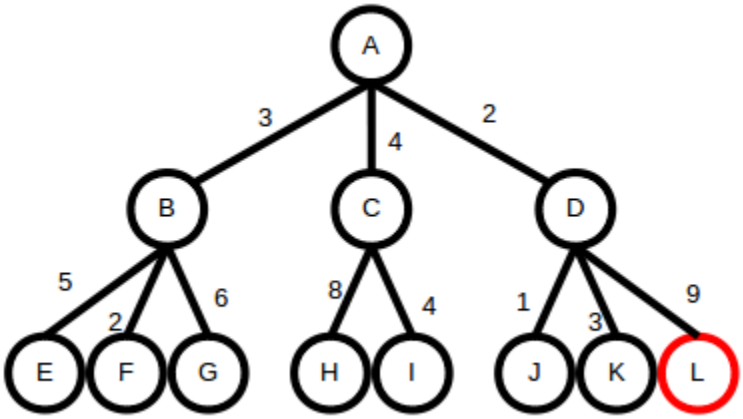
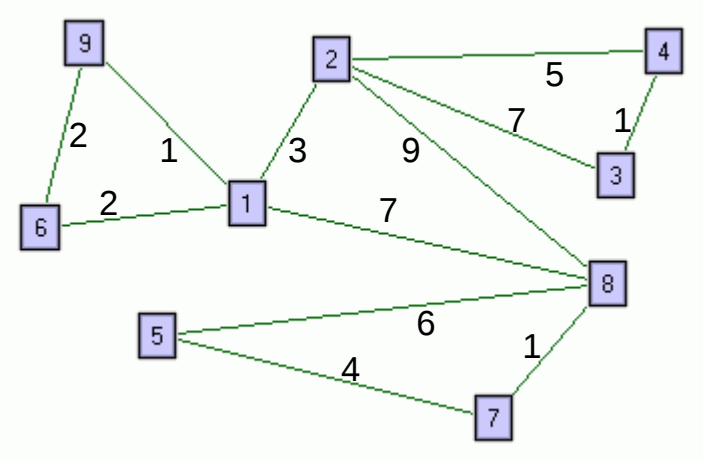
4511W, Fall-2018
ASSIGNMENT 2 :
**Assigned: 09/27/18 Due: 10/03/18 at 11:00 PM** (submit via moodle, you may scan or take a picture of your paper answers in a zip if you have multiple files). All assignments are individual.

**Problem 1**. (15 points)
Run iterative-deepening A* on the following tree. The initial state is A and the goal state is L.




| Node | Heuristic |
|---|---|
| A | 7 |
| B | 5 |
| C | 3 |
| D | 8 |
| E | 5 |
| F | 5 |
| G | 5 |
| H | 5 |
| I | 2 |
| J | 7 |
| K | 8 |
| L | 0 |

**Problem 2**. (20 points)

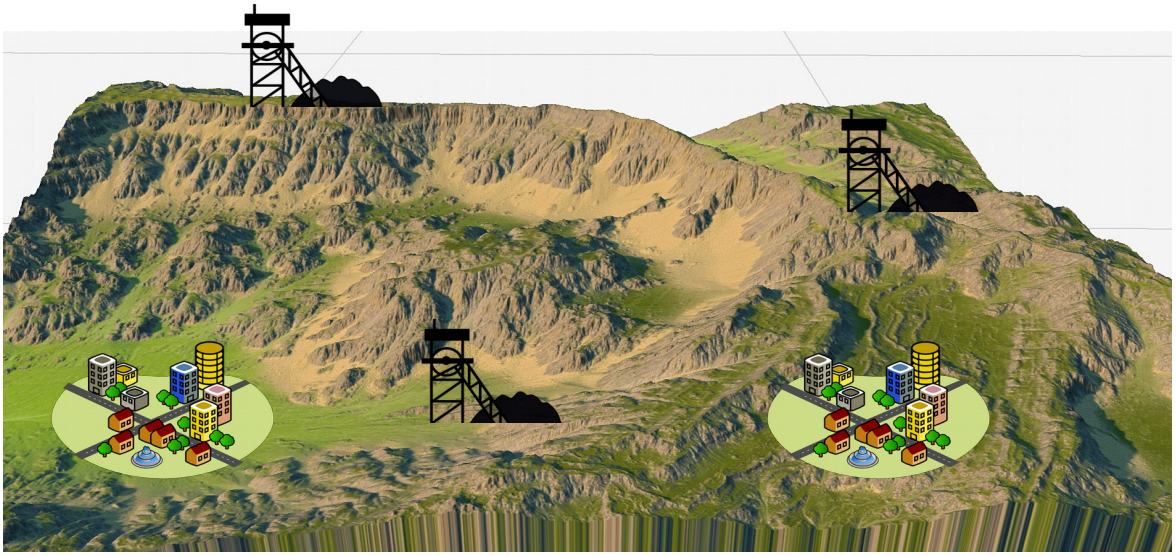



Heuristics:
Node 1 = 6
Node 2 = 4
Node 3 = 1
Node 4 = 0
Node 5 = 14
Node 6 = 10
Node 7 = 15
Node 8 = 15
Node 9 = 10

(1) Above is shown a graph and a heuristic where node #7 is the initial state and node #4 is the goal state. Is this heuristic admissible? Is it consistent? Clearly show your reasoning.

(2) Give an example of a graph with positive weights where uniform-cost-search will not find the goal in a finite amount of time (i.e. not be "complete").

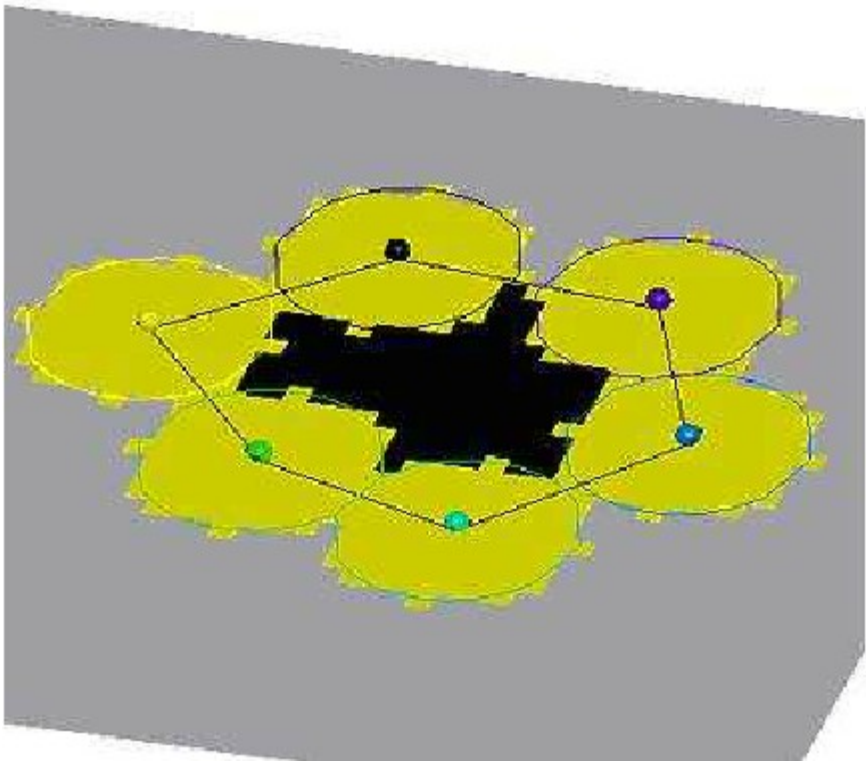
**Problem 3**. (30 points)
You are hired by a mining company to figure out how they should expand their mining operation. There are several locations where mines could be built along with several cities in the area. Factories can only be built in the city where the people live and each factory needs one mine "supplying" it. You also need to invest in building transport between the factories and mines as no such infrastructure is current there.



Each factory will give you a profit if it is operating, while each mine and road construction will have a cost. You can assume the engineers will tell you the lowest cost between any ~~factory/city~~ factory/mine you want to connect on this map. Your goal is to get the maximum profit.

For this problem, give: (1) an explanation how you will model this problem in the computer (2) a relaxation, (3) a general description of how to solve the relaxed optimally(without involving any search) and (4) how to compute the heuristic for a given state.

**Problem 4**. (20 points)
There is toxic spill that you need to clean up. The contaminated area is slowly expanding, so you need to station clean-up teams so that they can fully encompass the site to prevent the expansion. Each team has heavy equipment and cannot move, and also has a fixed radius they can work (all teams have the same radius). (Assume the environment is continuous, not in discrete grids as shown below. This means all circles must touch or overlap each other to count as "fully encompass".) The find the specific locations to surround the tocix spill with the least number of clean-up teams.



For this problem, give: (1) an explanation how you will model this problem in the computer (2) a relaxation, (3) a general description of how to solve the relaxed optimally(without involving any search) and (4) how to compute the heuristic for a given state.

## 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:

**8.736**u 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.