**CS 540: Introduction to Artificial Intelligence**
**Homework Assignment # 4**

**Assigned: 2/26**
**Due: 3/5 before class**

# Hand in your homework:

If a homework has programming questions, please hand in the Java program. If a homework has written questions, please hand in a PDF file. Regardless, please zip all your files into hwX.zip where X is the homework number. Go to UW Canvas, choose your CS540 course, choose Assignment, click on Homework X: this is where you submit your zip file.

# Late Policy:

All assignments are due at the beginning of class on the due date. One (1) day late, defined as a 24-hour period from the deadline (weekday or weekend), will result in 10% of the total points for the assignment deducted. So, for example, if a 100-point assignment is due on a Wednesday 9:30 a.m., and it is handed in between Wednesday 9:30 a.m. and Thursday 9:30 a.m., 10 points will be deducted. Two (2) days late, 25% off; three (3) days late, 50% off. No homework can be turned in more than three (3) days late. Written questions and program submission have the same deadline.

Assignment grading questions must be raised with the instructor within one week after the assignment is returned.

# Collaboration Policy:

You are to complete this assignment individually. However, you are encouraged to discuss the general algorithms and ideas with classmates, TAs, and instructor in order to help you answer the questions. You are also welcome to give each other examples that are not on the assignment in order to demonstrate how to solve problems. But we require you to:

- not explicitly tell each other the answers

- not to copy answers or code fragments from anyone or anywhere

- not to allow your answers to be copied

- not to get any code on the Web

In those cases where you work with one or more other people on the general discussion of the assignment and surrounding topics, we suggest that you specifically record on the assignment the names of the people you were in discussion with.

# Question 1: Inadmissible Heuristic Affects Completeness [50 points]

We saw in the class that if the heuristic function $h$ is inadmissible, A* search may find a suboptimal goal. We now show that A* may not even be complete, namely it may not terminate even if a goal with small cost exists. Consider the following graph where the right branch goes on forever: where $A$ is the initial state
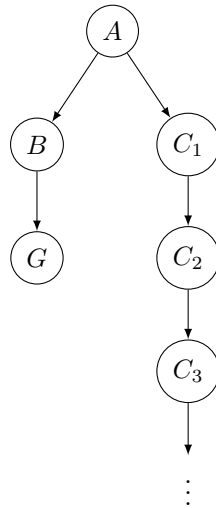


and $G$ is the only goal state. Let

$$cost(A, B) = 1/2, \quad cost(B, G) = 1/2$$

so the solution path from $A$ to $G$ has a cost of 1. Furthermore, let

$$cost(A, C_1) = 1 \quad cost(C_1, C_2) = 1/2 \quad cost(C_2, C_3) = 1/4 \quad cost(C_3, C_4) = 1/8 \quad \ldots$$

so that the costs decrease by half toward the right.

Suppose for all states $s$ the heuristic function $h(s) = 0$ *except* that $h(B) = 100$. This makes $h$ inadmissible. Nonetheless, let us run the A* algorithm with this $h$.

1. (5 points) By definition, what range of $h(B)$ is considered admissible?

2. (20 points) Run the A* algorithm (on slide 9 of part 2 of informed search) by hand for five iterations: that is, you should execute step 5 five times. At the end of *each* iteration, show the following:

   - states in OPEN
   - states in CLOSED
   - for each state, show its $f, g, h$ values
   - for each state, show its back pointer

3. (5 points) What is $\lim_{i \to \infty} f(C_i)$? Show your derivation.

4. (5 points) Explain in English why the search will not find $G$.

5. (10 points) Is there a range of $h(B)$ that is inadmissible but still allows A* search to find $G$? If yes, give the range; if no, explain why.

6. (5 points) Is admissible $h$ a sufficient condition, a necessary condition, both, or neither for A* search to find the optimal goal?

## Question 2: Simulated Annealing [25 points]

Consider a state space consisting of integers. Suppose that the successors of a state $x$ include all integers in $[x - 10, x + 10]$, including $x$ itself. Consider the score function $f(x) = \max\{4 - |x|, 2 - |x - 6|, 2 - |x + 6|\}$. It has three peaks with one forming a unique global maximum.

Recall that a random successor $y$ is generated on each iteration of simulated annealing. If this successor is better than the current state $x$, it is always accepted. If it is not better, it will still be accepted with probability

$$p = \exp\left\{-\frac{|f(x) - f(y)|}{T}\right\}.$$

Recall that to do something with probability $p$, you generate a random number $z \in [0, 1]$ uniformly and if $z \leq p$ you do it.

Use the temperature cooling scheme

$$T(i) = 2(0.9)^i,$$

where $i$ is the iteration number. E.g., for the first iteration, the temperature will be $T(1) = 2(0.9)^1 = 1.8$.

Perform 8 iterations of simulated annealing using $x = 2$ as your starting point. We already "randomly" picked a successor $y$ at each iteration for you, and also generated a "random number" $z$ at that iteration for you to decide whether to go to an inferior successor, should you need it. These are listed in Table 1. Use these numbers. For each iteration, write down (1) the current point, (2) the temperature (rounded to the nearest thousandth), and (3) the probability of moving to the successor given the successor and the temperature.

| Iteration Number | Random Successor $y$ | Random Number $z$ |
|:---:|:---:|:---:|
| 1 | 3 | 0.102 |
| 2 | 1 | 0.223 |
| 3 | 1 | 0.504 |
| 4 | 4 | 0.493 |
| 5 | 2 | 0.312 |
| 6 | 3 | 0.508 |
| 7 | 4 | 0.982 |
| 8 | 3 | 0.887 |

Table 1: **Successor States and Random Numbers**

# Question 3: City of Madison Open Data: Street Trees [25 points]

Visit the website `http://data-cityofmadison.opendata.arcgis.com`. Go City Datasets / SUSTAIN-ABILITY to find the "Street Trees" dataset:

```
Street Trees   thumbnail
Shared by CityOfMadison
The different classifications of the tree data are as follows: ID:...
Custom License 9/11/2017 Spatial Dataset 112,511 Rows
```

Once you get into this dataset's webpage, you can zoom in to the map on top of the webpage to see where the trees are. Try find Computer Science Department and see the few trees around the building! You will see many trees with their attributes. For your convenience, we have downloaded the dataset and provided it for this homework (available in both .csv and .txt format).

Suppose the city wants to inspect the trees for disease. An inspector starts from their office, drives a truck to visit each tree once, and goes back to the office at the end. The inspector wants to find the shortest route to do so. Because of traffic and one-way streets, the order the trees are visited affects the total distance the inspector needs to travel. It is therefore reasonable to represent a state by a permutation of the trees. Let there be $n$ trees. For a particular permutation $t_1, \ldots, t_n$, the total distance of that state is defined as

$$f(t_1, \ldots, t_n) = d(O, t_1) + \sum_{i=1}^{n-1} d(t_i, t_{i+1}) + d(t_n, O)$$

where $d(a, b)$ is the driving distance from $a$ to $b$, and $O$ represents the office. One can view $f$ as the score of the state, and the inspector wants to find a state with the minimum score.

1. How many states are there for $n$ trees?

2. The inspector can perform hill climbing to approximately minimize the score. To do so, a successor (neighborhood) function needs to be defined for each state. It turns out that one valid way to generate successors for the state $t_1, \ldots, t_n$ is to pick a position $j \in [1, n-1]$ and swap $t_j, t_{j+1}$. (If you are interested: any permutation can be expressed as a product of adjacent transpositions.) What fraction of the state space does one neighborhood cover? Note for this problem, the neighborhood of a state does not include the state itself.

3. Find $n$ from your Madison dataset and compute the number of states in scientific notation, round to one significant digit (i.e. $c \times 10^d$ where $c$ is a single digit). This is why the inspector cannot enumerate all states!

4. One way to estimate the worst case total distance is to assume that to travel between any two trees (or a tree and the office), the inspector has to travel the diameter of the city. Assume Madison has a diameter of 10 km. What is this worst case estimate of the total distance? Use the actual $n$ for Madison and express your answer in LD, namely the average earth moon distance. Keep one significant digit.

5. One way to estimate the best case total distance is to assume that to travel between any two trees (or a tree and the office), the inspector has to travel the minimum distance between any two trees. Assume that distance is 10 m. What is this best case estimate of the total distance? Express your answer in km.

6. Suppose the inspector drives at 25 miles per hour (note the Imperial unit). Can the inspector finish the job in one day?