CPSC 3750 – A.I.	**Due Feb. 26** (in class)
Assignment 2	Total marks: 70

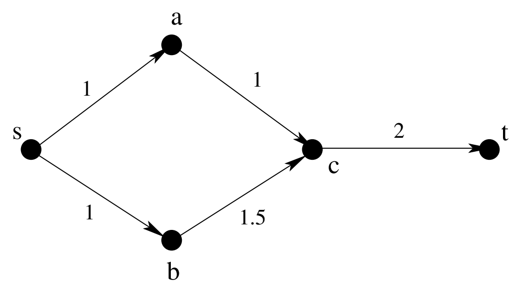
1) **(5 pts)** What is an admissible heuristic function? What is a consistent heuristic function?

   Solution:

   Admissible: does not over estimate the actual cost.

   Consistent: triangle inequality, the estimate from a state s is not larger than the estimate from a child of s plus the cost from s to that child. See also the text.

2) **(25 pts)** The figure below depicts a state space with five states. The successor function is shown by the oriented arcs and the initial and target states are s and t respectively. For example, from state s there are only two possible actions leading towards states a and b. Each edge is labelled with the cost of its corresponding action. For example, there are two possible paths leading to the target state, but the optimal path is *sact* with the path cost 4.



   (a) A trivial admissible heuristic function is given by the following values: $h(s) = h(t) = h(a) = h(b) = h(c) = 0$. Define another admissible heuristic function by listing its values.

   Solution:

   Another trivial admissible heuristic has value one for all states except t. Alternatively, a heuristic returning the optimal cost is also admissible.

   (b) As before, list the values of a consistent heuristic function.

   Solution:

   The above examples are also consistent.

   (c) List the values of a heuristic function that is admissible but NOT consistent.

   Solution:

   $h(b) = 2.9$, $h(b) = 3$, and $h(c) = 0.5$. Admisibility: $h(a) = 3 \leq 3$ (3 is the actual cost from a to t). Same conditions for b and c. However, $h(a) = 3 < h(c) + 1$ so $h()$ is not consistent.

   (d) Using the previous admissible but not consistent function, explain why A* search based on the *graph-search* algorithm from Fig. 3.19, p. 83 in your text (the search avoiding repeated states) could return a sub-optimal path, namely *sbct*.

   Solution:

   When a and b are added on the fringe, $f(b) < f(a)$ (recall that $f(x) = h(x) + d(s, x)$). Then c is expanded from b therefore c will never be tried from a (the optimal solution) because the search will not repeat states.

   (e) Using the heuristic function from (2c), trace the steps taken by the A* search based on the *tree-search* algorithm (which does not avoid repeated states). Draw a table where each row represents a step in the algorithm. For each row: list the fringe, give the objective function for each state in the fringe, and briefly explain the decision of the algorithm.

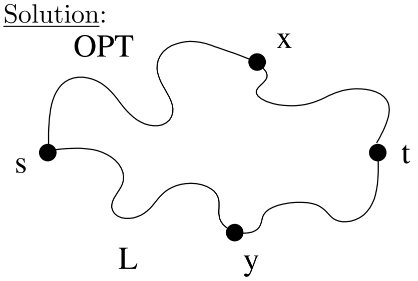
Solution:
Fringe states:

	Value of f			
Iterations	a	b	c	t
1	4	3.9		
2	4	-	3	
3	4	-	-	4.5
4	-	-	2.5	4.5
5	-	-	-	4

Note: the search repeates states and maintains the lowest cost among duplicate states.

3) **(10 pts)** Suppose you are given a non-admissible heuristic function that overestimates by at most a value c. Argue that A* search using this heuristic will never return a solution whose cost exceeds the optimal solution by more than c. If you cannot make a general argument, you may use the example from Problem 2

Solution:



Let L be the path that the A* algorithm returns and let OPT be the optimal path (see the figure). Let x be the deepest node from the optimal path discovered by A* right before A* returned L. Since L is returned, it means that $f(x) > f(t_L) = |L|$, where t_L is the target state following path L. But, since h overestimates by no more than c, $f(x) = d(s, x) + h(x) < d(s, x) + d(x, t) + c = |OPT| + c$. Therefore $L < OPT + c$.

4) **(5 pts)** Explain the difference between online and offline search agents.

Solution:
See the text. Online search is for partially observable environments.

5) **(10 pts)** Give a precise formulation as a constraint satisfaction problem for the *class scheduling problem.* Given are: a list of professors, a list of classes, a list of classrooms, and a list of possible time slots for the classes. Each professor has a list of classes that she can teach.

Solution:
We need to assign all classes. Let $c \in C$ be a class in the set C of all classes, and let p_c, t_c, and r_c be the variables for the prof, timeslot, and room assigned to class c. If the domains of variables are defined properly so that only capable profs teach classes, then we want to assign values to the variables to satisfy:

$$\forall c_1 \in C, c_2 \in C: \text{ if } t_{c_1} = t_{c_2} \text{ then } p_{c_1} \neq p_{c_2} \text{ and } r_{c_1} \neq r_{c_2}.$$

6) **(15 pts)** Solve the cryptarithmetic puzzle from Fig. 5.2 in your text, by hand, using forward checking with the minimum remaining values heuristic. Be brief and precise.

Solution:
Look at the constraints from the text. If we consider the remaining values, the first variables we try to assign are X_1, X_2 or X_3 as the carry can only be 0 or 1. Let's pick X_1.
Let $X_1 = 0$. Now, we pick X_3. We cannot assign 0 to X_3 because F would be 0 (forward checking) and it is the leading digit for the sum. So we pick 1.
Now, we choose F because its only legal value is 1. Let $F = 1$.
Pick X_2. Value 0 survives forward checking, so let $X_2 = 0$.
Next variable to assign is O because O must be even (constraint 3) and it is less than 5 (constraint 1), so remaining values are $\{0, 2, 4\}$. Forward checking on value 0 tells us it is not allowed (because R would also be forced to 0 and "Alldif" is violated. So choose 2, ... ETC.