**COMP 3710 Artificial Intelligence**

Fall 2018

Term test I

Student Name: Student Number:

1. (2 marks) There are two types of searching problems.
2. Explain how they are different.
3. For each type, give an example problem with an algorithm that can be used to solve the problem.
4. (5 marks) The Dijkstra’s algorithm is an example of A\* algorithm that uses heuristic\_ZERO. Given the following graph, shows how the algorithm works to find the goal node *G* from the start node *S*.

1

1

1

1

5

1

3

You need to show how the algorithm works by completing the following table. (Note that the number after each node name is the g-value. E.g., the g-value in (*X*, 6) is 6.

|  |  |  |
| --- | --- | --- |
| Visiting node | Queue – Expanded nodes | Queue – Visited nodes |
|  | (*S*, 0) |  |
| (*S*, 0) | (*A*, 1), (*B*, 1) | (*S*, 0) |
| (*A*, 1) |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. (2 marks) We would like to solve the 5-queens problem using a local search algorithm that moves 1 queen to another place so that the next board becomes better.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 5 |  | Image result for queen  symbol |  |  |  |
| 4 |  |  | Image result for queen  symbol |  |  |
| 3 | Image result for queen  symbol |  |  |  | Image result for queen  symbol |
| 2 |  |  |  | Image result for queen  symbol |  |
| 1 |  |  |  |  |  |

 a b c d e

Decide which queen can be move to which place. Justify your answer.

1. (2 marks) List the two heuristics used in CSPs. Explain how they are different.
2. (5 marks) We assume that the followings are valid.
* If *C* and *E*, then *F*
* If *F*, then *G*
* *C*
* *E*
1. Convert the above statements to BNFs.
2. Convert the above BNFs to CNFs.
3. Convert the above CNFs to a clause form
4. We would like to prove/disprove that *G* is valid. Use the resolution refutation. (You need to resolve one propositional symbol at each step.) You also need to interpret the resolution result.

1. (4 marks) Complete the following two functions that were used in the genetic algorithm for TSP.

function makeRouletteWheel(fitnesses)

{

 var wheel = [];

 for (var i = 0; i < fitnesses.length; i++)

 if (i == 0)

 wheel[0] = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

 else if (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

 wheel[i] = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

 else

 wheel[i] = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

 return \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

}

function selecTour(wheel)

{

 var r;

 r = Math.random();

 for (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

 if (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

 return \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

 return \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

}