**COMP 3710 Artificial Intelligence**

Winter 2018

Term test I

Student Name: Student Number:

1. (3 marks) There are two types of searching problems.
2. Explain how they are different.
3. For each type, give an algorithm, A\* or genetic algorithms, that can be used.
4. For each type, give an example problem.

|  |  |  |
| --- | --- | --- |
|  | Type 1 searching problem | Type 2 searching problem |
| a) | Goal is known.A solution is to find the shortest path to the goal from an initial node. | Goal is not known.A solution is to find a goal or similar one. |
| b) | A\* | Genetic algorithm |
| c) | n-Puzzle | Traveling salesman problem |

1. (3 marks) We would like to solve the 5-queens problem using Most-Constraining Variable First heuristic.

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

 a b c d e

You need to decide the next variable and its value, by completing the following table that shows the number of constraining cells in three columns. (The first and the second columns are already used.) For each variable, you should check only the rows that do not conflict with existing queens.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Row 1 | Row 2 | Row 3 | Row 4 | Row 5 |
| Variable c |  | 5 |  |  |  |
| Variable d | 4 | 6 |  | 6 |  |
| Variable e | 4 | 5 |  | 5 |  |

Decision: d:2 or d:4

1. (2 marks) Design a formal representation of problem state, i.e, a data structure with brief explanation for 8-Queens problem.
2. (4 marks) Answer the following questions about A\* algorithm for 3×3 puzzle game.
* The A\* algorithm uses the number of missed tiles heuristic.
* The goal board is [1, 2, 3, 4, 5, 6, 7, 8, 0].
* Node: (expanded board, g-value, h-value, parent board)
* The “selecting and visiting the next node” is explained in shaded rows.
* The “expansion of not-visited child nodes” is explained in non-shaded rows.

Complete the following table.

|  |  |  |
| --- | --- | --- |
| Visiting node | ExpandedQ | VisitedQ |
|  | { ([1,2,3,4,8,5,7,0,6], 0, 3, φ) } | {} |
| ([1,2,3,4,8,5,7,0,6], 0, 3, φ) | {} | { ([1,2,3,4,8,5,7,0,6], 0, 3, φ) } |
|  | { ([1,2,3,4,0,5,7,8,6], 1, 2, [1,2,3,4,8,5,7,0,6]),  ([1,2,3,4,8,5,0,7,6], 1, 4, [1,2,3,4,8,5,7,0,6]),  ([1,2,3,4,8,5,7,6,0], 1, 3, [1,2,3,4,8,5,7,0,6]) } | { ([1,2,3,4,8,5,7,0,6], 0, 3, φ) } |
| ([1,2,3,4,0,5,7,8,6], 1, 2, [1,2,3,4,8,5,7,0,6]) | { ([1,2,3,4,8,5,0,7,6], 1, 4, [1,2,3,4,8,5,7,0,6]),  ([1,2,3,4,8,5,7,6,0], 1, 3, [1,2,3,4,8,5,7,0,6]) } | { ([1,2,3,4,8,5,7,0,6], 0, 3, φ), ([1,2,3,4,0,5,7,8,6], 1, 2, [1,2,3,4,8,5,7,0,6]) } |
|  | { ([1,2,3,4,8,5,0,7,6], 1, 4, [1,2,3,4,8,5,7,0,6]),  ([1,2,3,4,8,5,7,6,0], 1, 3, [1,2,3,4,8,5,7,0,6]), ([1,0,3,4,2,5,7,8,6], 2, 3, [1,2,3,4,0,5,7,8,6]), ([1,2,3,4,5,0,7,8,6], 2, 1, [1,2,3,4,0,5,7,8,6]), ([1,2,3,0,4,5,7,8,6], 2, 3, [1,2,3,4,0,5,7,8,6]), } | { ([1,2,3,4,8,5,7,0,6], 0, 3, φ), ([1,2,3,4,0,5,7,8,6], 1, 2, [1,2,3,4,8,5,7,0,6]) } |
| ([1,2,3,4,5,0,7,8,6], 2, 1, [1,2,3,4,0,5,7,8,6]) | { ([1,2,3,4,8,5,0,7,6], 1, 4, [1,2,3,4,8,5,7,0,6]),  ([1,2,3,4,8,5,7,6,0], 1, 3, [1,2,3,4,8,5,7,0,6]), ([1,0,3,4,2,5,7,8,6], 2, 3, [1,2,3,4,0,5,7,8,6]), ([1,2,3,0,4,5,7,8,6], 2, 3, [1,2,3,4,0,5,7,8,6]), } | { ([1,2,3,4,8,5,7,0,6], 0, 3, φ), ([1,2,3,4,0,5,7,8,6], 1, 2, [1,2,3,4,8,5,7,0,6]), ([1,2,3,4,5,0,7,8,6], 2, 1, [1,2,3,4,0,5,7,8,6]) } |

1. (2 marks) Complete the following table about genetic algorithms, that shows which components support ‘cumulative improvement’ and ‘diversity’. You just need to mark related components.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Multiple individual in each generation | Selection based on fitness | Crossover | Mutation |
| Cumulative improvement |  | x | x |  |
| Diversity | x |  | x | x |

1. (5 marks) The followings are valid.
* *C* → *D*
* *D* ∧ *B* → *E*
* *C*

You need to show whether *E* is valid or not, using the resolution refutation.

1. Convert the above BNFs to CNFs.

~C ∨ D

~(D ∧ B) ∨ E ≅ ~D ∨ ~B ∨ E

C

1. Covert the CNFs to clause form

{ (~C, D), (~D, ~B, E), C }

1. Resolve the clause form step-by-step. (You need to resolve one propositional symbol at each step.)

{ (~C, D), (~D, ~B, E), C, ~E }

* { D, (~D, ~B, E), ~E }
* { (~B, E), ~E }
* { ~B }
1. Interpret the resolution result.

The validity of E is dependent on B.

* B: valid => ~E: invalid => E: valid
* B: invalid => ~E: valid => E: invalid