Name: ____________________________ **Digipen login:** ____________________________

Part II - open book/notes

You have to submit Part I BEFORE proceeding with Part II

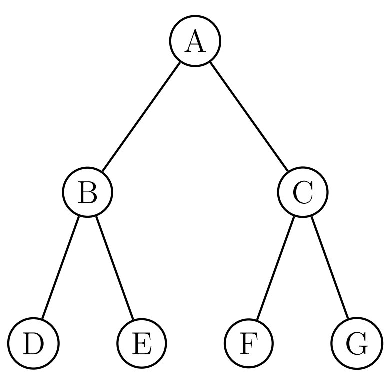
1. **Problem** (7 pts): All versions of search algorithm are special cases of a general algorithm TREE-SEARCH. Write it – use the following operations as given/implemented: ISEMPTY, INSERT_ALL, POP_FRONT, EXPAND, ISGOAL

   (a) Which operation of TREE-SEARCH makes BFS,DFS,A*, etc different?

   (b) How exactly the operation from the previous question is implemented in DFS and BFS.

   (c) When does TREE-SEARCH return a failure? (add it to your algorithm if you haven't done it already).

2. **Problem** (3 * 2 pts): Specify the order of states in which they are expanded (processed) by BFS and DFS.



3. **Problem** (3+3+3 pts): Which property of search algorithm is referred to as **completeness**.

   Which property of search algorithm is referred to as **optimality**.

   Is there a relationship between the two?

4. **Problem** (1+1+2+1+1+2 pts): Answer each of the question with a single expression/word: use the following notations d - solution depth, b - maximum branching factor, m - maximum depth of the tree.

   - What is time complexity of breadth-first search.
   - What is space complexity of breadth-first search.
   - Is breadth-first search optimal?
   - What is time complexity of depth-first search.
   - What is space complexity of depth-first search.
   - Is depth-first search optimal?

5. **Problem** (3+3+3+4 pts): Define **admissible** heuristic.

   Define **consistent** heuristic.

   Is there a relationship between the two?

   Why is it important that heuristic is admissible?

6. **Problem** (5 pts): Consider A* with heuristic equal 0 for all states. Which of the known seacrh algorithms is it identical to?

7. **Problem** (8 pts): Given directed graph ($A \to B, A \to C, C \to B, B \to D, C \to D$) draw the corresponding search tree.

8. **Problem** (12 pts): Consider a domain with an agent on a straight line (1-D). Agent is to perform an A* from 0 to 5 to optimize **time** (i.e. fastest time). The agent is able to

   (a) walk at a speed 1 unit/link per second (any time even after running)

   (b) rest for 1 second and then run at a speed 2 units per second for a maximum 2 seconds

   Draw a search tree (4 levels) for this problem – mark each state with 2 numbers

   (a) location 0,...,5

   (b) g-value

   And each link with an action it corresponds to (**r** – run, **s** – sleep/rest, **w** – walk)

   Explain heuristic – comment whether it's admissible and consistent.

   Extra (8 pts) Solve problem by A*. Show steps and the content of the openlist.

9. **Problem** (12 pts): Consider the standard hill-climbing (local search) algorithm without random restart applied to the following problem:
Domain: sequences of 6 bits.
Fitness function:

$$f = (\text{number of zeros in the sequence}) + 3 * (\text{number of sequences of 3 consequtive 1s})$$

so that

$$f(110011) = 2 + 3 \times 0 = 2$$

$$f(111000) = 3 + 3 \times 1 = 6$$

$$f(111100) = 2 + 3 \times 2 = 8$$

in the last example sequence 1111 contains 2 triples 111's.

$$f(111111) = 0 + 3 \times 4 = 12$$

Neighbor relation: two sequences are neighbors if they differ in a single bit only.
Starting point: algorithm starts with sequence 100100.

Do you see any problems? If yes - **explain** it, try to use the correct terminology.

Which algorithm covered in class may solve the problem. **Explain** why. You are NOT ALLOWED to modify neighbor relation and/or starting point.

10. **Problem** (15 pts): Consider an agent in 2-D space on a square grid (agent can move in 4 basic directions only). Assume agent is located on a vertex of the grid (rather then inside a square). Agent is performing an A* search to optimize distance (pathfinding) from $(0, 0)$ to $(10, 10)$ (which is 20 steps). How many iterations of A* algorithm (so this is **time**-related question) are required to solve this problem using

    (a) A* with Manhattan heuristic

    (b) A* with Euclidean heuristic

    (c) BFS

    (d) DFS

    (e) greedy best-first ($f(x) = h(x)$)

    Reminder: during each iteration search algorithm removes a single node from the openlist and expands it.

    Hint: if there is no single answer – write a sentence explaining the possibilities.

    Hint: if you cannot calculate the number – draw it.

    Hint: in the textbook p. 100 there is a paragraph saying:

    If C^* is the cost of the optimal solution, then we can say the following:

    - A^* expands all nodes with $f(n) < C^*$
    - A^* expands some nodes with $f(n) = C^*$
    - A^* expands no nodes with $f(n) > C^*$

    Note: if your answer is a product of several numbers – just write it as a product, i.e. I' prefer 123×321 over 39483

    What's going to happen in this particular example if A* uses heuristic which is Manhattan distance to goal multiplied by a 1.001. Those who took cs380 may remember that this technique may speed up A*. Explain why, be specific.

$$h(x) = (1 + \epsilon) \times Manhattan(x, Goal)$$