Q1

Circle (O) the true statements, and cross (X) the false statements.

1. Breadth-first search is a special form of uniform-cost search where the cost $g(n)$ is the number of steps from the initial state to state $n$.
2. Depth-first tree search is complete if the state space is finite.
3. Backtracking search is a variant of depth-first search which generates only one successor at a time rather than all successors.
4. A heuristic function $h(n)$ is said to be admissible if it never exceeds the real optimal cost from $n$ to any goal state. Therefore, $h(n) = 0$ is always admissible.
5. Uniform-cost search is a special form of A* search.
This question refers to the undirected weighted graph below:

![Graph Image]

A is the start state and G is the goal state. The costs are given as weights on the graph. Apply each of the following algorithms until 5 expansions are made or a solution is found, and show what are in frontier and what are in explored. It is assumed that the children of each node are ordered alphabetically, and backtracking search generates nodes according to the order.

1. Breadth-first graph search.

2. Backtracking graph search.

3. A* graph search with the following heuristic function:

<table>
<thead>
<tr>
<th>n</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>h(n)</td>
<td>9</td>
<td>9</td>
<td>8</td>
<td>7</td>
<td>1</td>
<td>4</td>
<td>0</td>
</tr>
</tbody>
</table>
Q3

Consider the \textit{pigeonhole} problem. Given \( n \) pigeons and \( m \) holes, each of which can hold \( k \) pigeons, the goal of the problem is to put the pigeons into the holes such that every pigeon is assigned a hole and no more than \( k \) pigeons are put into any hole. Obviously, according to the pigeon-hole principle, if \( n > m \times k \), then the problem is unsatisfiable.

1. Model the problem as a CSP.
2. The original problem permits symmetric solutions. Introduce symmetry-breaking constraints into your model so that some of the symmetries are eliminated.
3. (Extra 5 points) Implement your model in a programming language of your choice.
Q4

Consider the relaxed 8-puzzle in which a tile can move from any square to the empty square.

- Formulate the problem as a state-space search problem by giving a state representation, a goal test, a set of actions, and a heuristic function.

- (Extra 5 points) Implement your model in a programming language of your choice for the following problem instance:

```
Start State

```

```
Goal State

```