Answers to some practice complexity questions

1. The smallest complexity class to which the function $T(n) = n^2 \times (n + 1)$ belongs is:

   (a) $O(n)$
   (b) $O(n + 1)$
   (c) $O(n^2)$
   (d) $O(n^3)$
   (e) $O(2^n)$

   Answer: (d)

2. Time complexity of breadth-first search (V is the number of vertices, E the number of edges) is in

   (a) $O(V)$
   (b) $O(V + E)$
   (c) $O(E)$
   (d) $O(V^2)$
   (e) $O(E^2)$

   Answer: (c)

3. What is the worst case time complexity of the search algorithm below (where n is the length of the array):

```
boolean search(int[] array, int value){
   for (int j = 0; j < array.length; j++) {
      if (array[j] == value) return true;
   }
   return false;
}
```

   (a) $O(1)$
   (b) $O(log_2 n)$
   (c) $O(n)$
   (d) $O(n^2)$
   (e) $O(n + value)$

   Answer: (c)

4. An input to an algorithm is a model with n states. The algorithm checks every pair of states, so it makes n^2 steps. Is it polynomial or exponential in n?

   Answer: polynomial

5. An input to an algorithm is a model with n states, k agents, d actions, and m transitions. The algorithm traverses each transition once for every state, so it makes nm steps. Is it polynomial or exponential in n and m?

   Answer: polynomial in both

6. An input to an algorithm is a model with n states, k agents, d actions, and m transitions. The algorithm generates all possible memoryless strategies for each of the agents. A memoryless strategy is an assignment of a single action to each state.

   (a) assuming each agent has d actions in each state, how many different memoryless strategies are there for one agent?

   Answer: d^n (d choices in the first state, times d choices in the second states, ... So $d \times d \times \cdots \times d$, n times.)

   (b) how many strategies does the algorithm generate? Is this number polynomial or exponential in k? Is this number polynomial or exponential in n?

   Answer: $k \times d^n$. This is polynomial in k but exponential in n.

7. An input to an algorithm is a model with n states, k agents, d actions, and m transitions, and a state q in the model. The algorithm generates all possible choices of actions in q for each agent, a set of pair $(agent, action)$. How many pairs does the algorithm generate? Is this number exponential in any of the parameters n, k, d, m?

   Answer: if each agent has d actions in q, then there are d choices for each agent, so $k \times d$ pairs. This is polynomial in k and d.

8. An input to an algorithm is a model with n states, k agents, d actions, and m transitions, a state q in the model, an agent i, and a proposition p. The algorithm needs to check if the agent has a strategy to enforce p in the next state. To check whether i can enforce p in the next state, how many actions do we need to check? Can this check be done in time polynomial in m? (Assume that checking whether a state at the end of a transition satisfies p can be done in constant time.)

   Answer: we need to check at most d actions. The check involves going through all transitions out of q and for each of d actions checking whether all transitions of which this action is part of, the resulting state satisfies p. There are at most m transitions out of q. A straightforward approach (not the most efficient) is to iterate over d actions, and for each action, iterate over m transitions. Even this is $d \times m$ so polynomial.