Please remember to justify your answers carefully.

Last name: __________________________ First name: __________________________
Question 1 (Probability and Bayes’ rule.)
Consider two events. Under Event 1 ($E_1$), a student was born in Antarctica, and its probability is $\Pr(E_1) = 0.3$. Under Event 2 ($E_2$), the student has a purple test, and its probability is $\Pr(E_2) = 0.4$. Moreover, the probability that the student was born in Antarctica yet has a non-purple test is $\Pr(E_1, (E_2)^C) = 0.2$, where $(\cdot)^C$ denotes the complement of a set or event.

(a) Compute $\Pr(E_1)$, $\Pr((E_1)^C)$, $\Pr(E_2)$, $\Pr((E_2)^C)$, $\Pr(E_1, E_2)$, $\Pr(E_1, (E_2)^C)$, $\Pr((E_1)^C, E_2)$, and $\Pr((E_1)^C, (E_2)^C)$.

(b) Given that the student has a purple test, what is the probability that they were born in Antarctica? (Hint: use Bayes’ rule.)
Question 2 (MDL and overfitting.)
Wikipedia states the following about overfitting.

In statistics, overfitting is "the production of an analysis that corresponds too closely or exactly to a particular set of data, and may therefore fail to fit additional data or predict future observations reliably".

In the context of the minimum description length (MDL) principle, consider a model class $C$ comprised of $M$ models, $C_1, C_2, \ldots, C_M$, where the simpler models are numbered with lower indices ($C_1$ is simplest and $C_M$ most complicated). From an overfitting perspective, suppose that as we move toward complicated models, they perform better on training data, but results on test data become messy beyond some "reasonable" model, $C_j$.

Based on the overfitting information provided, compare the coding lengths you expect to be required for $C_1$, $C_j$, and $C_M$. (Which is smallest, and why?) Make sure to justify your answer.
Question 3 (Computational complexity.)
Consider the factorial function, \( N! = \prod_{n=1}^{N} n \).

(a) Show that \( N! = O(N^N) \).

(b) Show that \( N! = \Omega(2^N) \).
**Question 4** (Algorithms.)

Develop an algorithm that processes a sorted list of different positive integers, and returns the smallest number missing from the list. For example, below we see input lists $x$ and outputs $y$ for several examples.

<table>
<thead>
<tr>
<th>Input $x$</th>
<th>[1, 2, 5]</th>
<th>[2, 3, 4]</th>
<th>[1, 2]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output $y$</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>

While answering the question, please denote the length of $x$ by $N$, and the $n$th number in the list by $x(n)$. Ideally, your algorithm should be efficient. However, if its computational complexity is greater than ours, you will still get most of the credit.

(a) Express your algorithm in words or pseudocode. (No need to deal with syntax details of any programming language.)

(b) What is the computational complexity of your algorithm? Make sure to justify your answer.
Question 5 (Newton’s method.)
Consider the function, \( f(x) = x^4 - 4 \). We want to find its minimum. It can be shown that the minimum occurs at \( x_{\text{min}} = 0 \), and \( f_{\text{min}} = f(x_{\text{min}}) = -4 \). However, in general we will often try to optimize more complicated functions. In this question, you will compute the minimum using Newton’s method.

(a) Finding the minimum of \( f(x) = x^4 - 4 \) is analogous to finding a root of its derivative. Compute the derivative, \( f'(x) \).

(b) To find a root of \( f'(x) \), we will begin in iteration \( t = 0 \) with an initial guess, \( x_0 \). In iteration \( t \), we will update our guess as follows,

\[
x_{t+1} = x_t - \frac{f'(x_t)}{f''(x_t)},
\]

where the first Newton iteration computes \( x_1 \) from \( x_0 \). Compute the second derivative, \( f''(x) \).

(c) Let’s initialize our algorithm with \( x_0 = 3 \). Please compute \( x_1 \) using a Newton iteration (1) and your results for the derivatives. (If you are not sure about \( f'(x) \) and \( f''(x) \) from Parts a and b, then you can assume that \( f'(x) = x^2 \) and \( f''(x) = 3x \).)