Please read the following guidelines carefully.

- Please print your name, student number and UTORid on the front of the test.
- This test has 4 questions. There are a total of 8 pages, DOUBLE-SIDED.
- Answer questions clearly and completely. Provide justification unless explicitly asked not to.
- In your proofs, you may use definitions of predicates presented in the course. You may not use any external facts about rates of growth, divisibility, primes, or greatest common divisor unless you prove them, or they are given to you in the question.
- For algorithm analysis questions, you can jump immediately from a step count to an asymptotic bound without formal proof (e.g., you may write “the number of steps is at most $3n + \log n$, which is $O(n)$”).

Take a deep breath.

This is your chance to show us

How much you’ve learned.

We WANT to give you the credit

That you’ve earned.

A number does not define you.

<table>
<thead>
<tr>
<th>Question</th>
<th>Grade</th>
<th>Out of</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Q2</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Q3</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>Q4</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>20</strong></td>
<td></td>
</tr>
</tbody>
</table>
1. [3 marks] Various topics.

(a) [1 mark] **Number representations.** Put an “X” in the box next to the base 2 representation of the decimal number 10. (No justification is required.)

**Hint:** \(2^0 = 1, 2^1 = 2, 2^2 = 4, 2^3 = 8, 2^4 = 16, 2^5 = 32\).

- [ ] 110
- [ ] 1010
- [ ] 1001
- [ ] 10010

(b) [1 mark] **Asymptotic descriptions.** Consider the function \(f(n) = 5n^2 - 6n + 7\), where \(n \in \mathbb{N}\). Put an “X” in every box that is beside a statement that is True. (No justification is required.)

- [ ] \(f(n) \in O(n)\)
- [ ] \(f(n) \in \Omega(1)\)
- [ ] \(f(n) \in \Omega(n^2)\)
- [ ] \(f(n) \in O(n^3)\)

(c) [1 mark] **Choosing the base case.** Consider the statement:

\[
\forall n \in \mathbb{N}, n \geq n_0 \Rightarrow (2n + 1) \leq 2^n
\]

Put an “X” in the box next to the smallest base case number \(n_0\) for which the statement is True. (No justification is required.)

- [ ] 0
- [ ] 1
- [ ] 2
- [ ] 3
2. [5 marks] **Induction.** Prove the following statement using induction on $n$:

$$
\forall n \in \mathbb{N}, \exists k \in \mathbb{N}, \ 5k = 6^n + 4
$$
3. [6 marks] Worst-case running time. Consider the following algorithm, which takes as input a list of positive numbers.

```python
def alg(A):
    m = len(A)
    count = 0
    i = 1
    j = m
    while i < 2^m and j > 1:  # 2^m means "2 to the power of m."
        if count < i:
            count = count + A[i]
            i = 2 * i
        else:
            j = j // 2  # Integer division; rounds down.
    print('Count exceeded limit')
```

Prove that the worst-case running time of the above algorithm is \( \Theta(m) \), where \( m \) is the length of the input list. Note that this requires two proofs: that the worst-case runtime is \( O(m) \), and that the worst-case runtime is \( \Omega(m) \). Be sure to state exactly what you’re proving in each part of your solution.

Guidelines: you can assume that any line or block of code within the loop takes constant time. To save time, you do not need to use ceilings and floor function to round your expressions.

Note: If you run out of space on this page, continue your solution on the next page.
You may continue your solution to Question 3 on this page.
Question 4 is on the back of this page.
4. [6 marks] **Best-case running time.** Prove that the best-case running time of the algorithm from Question 3 is $\Theta(\log m)$, where $m$ is the length of the input list. Note that this again requires two proofs, similar to the previous question. The same guidelines from that question apply here as well.