HOMEWORK 3
COMPUTATIONAL COMPLEXITY *

10-607 COMPUTATIONAL FOUNDATIONS FOR MACHINE LEARNING

START HERE: Instructions

- **Collaboration Policy**: Please read the collaboration policy in the syllabus.
- **Late Submission Policy:** See the late submission policy in the syllabus.
- **Submitting your work:** You will use Gradescope to submit answers to all questions.
  - **Written:** For written problems such as short answer, multiple choice, derivations, proofs, or plots, please use the provided template. Submissions can be handwritten onto the template, but should be labeled and clearly legible. If your writing is not legible, you will not be awarded marks. Alternatively, submissions can be written in LaTeX. Each derivation/proof should be completed in the boxes provided. To receive full credit, you are responsible for ensuring that your submission contains exactly the same number of pages and the same alignment as our PDF template.
  - **Latex Template:** https://www.overleaf.com/read/nvhbhvgqztpb#02928c

Question	Points
Big-O	5
Proving Big-O Runtimes	12
Counting Operations	6
Total:	23

[*] Compiled on Monday 3rd November, 2025 at 07:40

Instructions for Specific Problem Types

For "Select One" questions, please fill in the appropriate bubble completely:

Select One: Who taught this course?

● Bryan Wilder

○ Marie Curie

○ Noam Chomsky

If you need to change your answer, you may cross out the previous answer and bubble in the new answer:

Select One: Who taught this course?

● Bryan Wilder

○ Marie Curie

⊗ Noam Chomsky

For "Select all that apply" questions, please fill in all appropriate squares completely:

Select all that apply: Which are scientists?

■ Stephen Hawking

■ Albert Einstein

■ Isaac Newton

□ I don't know

Again, if you need to change your answer, you may cross out the previous answer(s) and bubble in the new answer(s):

Select all that apply: Which are scientists?

■ Stephen Hawking

■ Albert Einstein

■ Isaac Newton

□ I don't know

For questions where you must fill in a blank, please make sure your final answer is fully included in the given space. You may cross out answers or parts of answers, but the final answer must still be within the given space.

Fill in the blank: What is the course number?

10-606

10-60~~6~~7

1 Big-O (5 points)

1. (1 point) **True or False:** $n^2 \in O(n^2)$

   ○ True

   ○ False

2. (1 point) **True or False:** $3n + n\log(n) + n^2 \in O(n\log(n))$

   ○ True

   ○ False

3. (1 point) **True or False:** $n^{100} \in O(\frac{1}{101}2^n)$

   ○ True

   ○ False

4. (1 point) **True or False:** If $f(n) \in O(n^4)$, then it must be the case that $f(n) \in O(n^5)$.

   ○ True

   ○ False

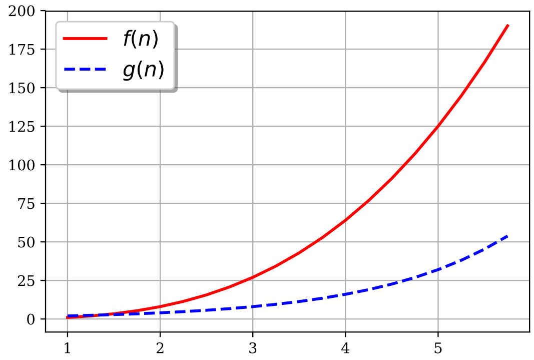

Figure 1

5. (1 point) Suppose there exist two functions whose range and domain are the real numbers $f(n) : \mathbb{R} \to \mathbb{R}$ and $g(n) : \mathbb{R} \to \mathbb{R}$, but the only information you have about them is the plot in Figure 1 showing the range $[1, 6]$. **True or False:** For any function h such that $f(n) \in O(h(n))$, it must be that $g(n) \in O(h(n))$.

   ○ True

   ○ False

2 Proving Big-O Runtimes (12 points)

1. (4 points) Prove or disprove that $4n^4 + 3n^3 + 2n^2 + n \log n$ is in $O(n^4)$?

   Each statement in your proof should have a justification, however, you don't need to explicitly write this justification. Make sure to avoid steps in your proof that are too large to be justified.

   Proof

2. (4 points) Prove or disprove that $\log n$ is in $O((\log(n))^2)$.

   Each statement in your proof should have a justification, however, you don't need to explicitly write this justification. Make sure to avoid steps in your proof that are too large to be justified.

Proof

3. (4 points) Use Induction to prove or disprove that 2^n is in $O(n!)$.

Proof

3 Counting Operations (6 points)

1. (2 points) What is the runtime complexity of the following code in terms of n. Give the simplest and tightest bound.

```
def f(n):
    result = 0
    j = 0
    while j < n
        k = 1
        while k<n:
            result += 1
            k *= 2
        j++
```

Complexity

2. (2 points) What is the runtime complexity of the following code in terms of n. Give the simplest and tightest bound. You can assume that `arr` is an array of floating point numbers and that indexing into it (e.g. `arr[k]`) takes constant time.

```
def g(arr):
    n = len(arr)
    ms = float(’-inf’)
    for i in range(n):
        for j in range(i, n):
            cs = 0
            for k in range(i, j + 1):
                cs += arr[k]
            ms = max(ms, cs)
    return ms
```

Complexity

3. (2 points) What is the runtime complexity of the following code in terms of n. Give the simplest and tightest bound.

```
def h(n):
    i = 0
    while i < n:
        j = 0
        while j <= i:
            j += (i + 1)
        i += 1
```

Complexity

4 Collaboration Questions

After you have completed all other components of this assignment, report your answers to these questions regarding the collaboration policy. Details of the policy can be found in the syllabus.

1. Did you receive any help whatsoever from anyone in solving this assignment? If so, include full details.
2. Did you give any help whatsoever to anyone in solving this assignment? If so, include full details.
3. Did you find or come across code that implements any part of this assignment? If so, include full details.

Your Answer