Name: Yogendra Pokhrel

Student ID: 617585

Math Problem 1:Increasing and Eventually Nondecreasing

A function is increasing if its derivative is always positive ($f'(x)>0f'(x) > 0$).
A function is eventually nondecreasing if its derivative is nonnegative ($f'(x)\geq 0f'(x)$ \geq 0) for large xx.

(1) $f(x)= -x^2$

- The derivative is: $f'(x)= -2x$
- This is negative for all $x\neq 0$, meaning the function is always decreasing.
- Since it never increases, it is not increasing and not eventually nondecreasing.

(2) $f(x) = x^2 + 2x + 1$

- The derivative is: $f'(x) = 2x + 2$
- This is positive for $x > -1$, meaning the function increases for $x > -1$.
- However, for $x < -1$, $f'(x) < 0$, so it decreases in that region.
- Since it eventually increases for large xx, it is not always increasing but eventually nondecreasing.

(3) $f(x) = x^3 + x$

- The derivative is: $f'(x) = 3x^2 + 1$
- Since $3x^2+ 1$ is always positive for all x, the function never decreases.
- Therefore, f(x) is increasing everywhere and also eventually nondecreasing.

Math Problem 2: Asymptotic Growth Comparison

We compare the growth rates of f(x) and g(x) using **Big-O notation**.

- If f(x) is O(g(x)), then f(x) grows no faster than g(x).
- If g(x) is O(f(x)), then g(x) grows no faster than f(x).
- If both hold, they grow at the same rate.

(1) $f(x) = 2x^2$, $g(x) = x^2 + 1$

- For large x, the +1 in g(x) becomes insignificant, so g(x) is approx x^2.
- Since $2x^2$ and $x^2 + 1$ both grow at the rate of x^2, they have the same asymptotic growth: $f(x)=O(g(x))$ and $g(x)=O(f(x))$
- Conclusion: Both grow at the same rate.

(2) $f(x) = x^2$, $g(x)= x^3$

- As x is infinity, x^3 dominates x^2.
- That means x^2 grows slower than x^3
    - $f(x)=O(g(x))$,but $g(x)\neq O(f(x))$
- Conclusion**:** f(x) grows slower than g(x).

(3) $f(x) = 4x + 1$, $g(x) = x^2 - 1$

- For large x, $x^2 - 1$ behaves like x^2.
- Since x^2 grows faster than 4x, we have: $f(x)=O(g(x))$,but $g(x)\neq O(f(x))$
- Conclusion**:** f(x) grows slower than g(x).

Problem 1 & 2: Java Algorithm Solution in the same directory

Problem 3: Greedy Strategy for Subset Sum Problem

The given greedy strategy for solving the Subset Sum Problem does not always work. While it may find the correct subset in some cases, there are situations where it fails to identify the correct subset that sums to the given target value.

For example

Let's consider the following set and target sum:

S = {1, 4, 6, 8}
k = 10

Step 1: Sorting the Set
The sorted set remains: S = {1, 4, 6, 8}

Step 2: Applying the Greedy Algorithm

- Start with an empty set T.
- Add 1 to T since 1 is less than or equal to k. Now, T = {1}, sum = 1.
- Add 4 to T since 1 + 4 = 5 is less than or equal to k. Now, T = {1, 4}, sum = 5.
- Try adding 6 to T, but 1 + 4 + 6 = 11, which exceeds k. So, 6 is not added.
- Try adding 8 to T, but 1 + 4 + 8 = 13, which exceeds k. So, 8 is not added.

The algorithm returns T = {1, 4} with a sum of 5, which is incorrect.

Step 3: Finding the Correct Subset
The correct subset that sums to 10 is {4, 6}, but the greedy approach does not find it because it makes decisions based on the smallest numbers first, leading to an incorrect result.

The greedy strategy makes local optimal choices at each step but does not guarantee a globally optimal solution. It assumes that taking the smallest numbers first will always lead to the correct subset, but in some cases, skipping a smaller number allows for finding the correct sum which is why it fails sometimes.

Problem 4: Removing the Last Element in Subset Sum Problem

Yes, if T is a correct solution to the Subset Sum problem and it includes s_{n-1}, then removing sn-1 from T will still be a correct solution for a new problem where:

- S0 = $\{s_0, s_1, ..., s_{n-2}\}$ (the original set without s_{n-1})
- k0 = k - s_{n-1} (the new target after removing s_{n-1})

Why is this true?

- Since T is a solution, the sum of all elements in T is exactly k.
- If s_{n-1} is in T, removing it decreases the sum by s_{n-1}.
- The remaining numbers in T must add up to k - s_{n-1}, which is exactly what we need for the new problem (S', k').

For an example

Let S = {1, 4, 6, 9} and k = 10.
A valid solution is T = {4, 6}, because 4 + 6 = 10.

Now, if we remove sn-1 = 6, we get:

- S0 = {1, 4, 9}
- k0 = 10 - 6 = 4
- The remaining subset {4} in T still sums to 4, so it is a correct solution for the new problem.

Therefore, Yes, if a subset T includes s_{n-1} and is a valid solution, then after removing s_{n-1}, the remaining numbers still solve the problem for the new target k - s_{n-1}.