CSE 3500 – Algorithms and Complexity
Homework 1

(30 points) Question 1

Consider the following linear search algorithm that searches for a target element within an array of numbers.

```
Function search(array of size n, target):
    for each element in array:
        if element == target:
            return "Element found"
    return "Element not found"
```

Analyze the best-case, average-case, and worst-case scenarios for this linear search algorithm based on the position of the target element within the array.

a) **Best Case:** What is the best-case scenario for this linear search algorithm? Explain the situation where it would perform most efficiently.
b) **Average Case:** How does the average-case scenario consider different possible positions of the target element within the array? Provide an explanation of how the algorithm performs on average.
c) **Worst Case:** What is the worst-case scenario for this linear search algorithm? Describe the input arrangement that would lead to the highest time complexity.

(25 points) Question 2

Consider a scenario where you need to find the sum of all even numbers between 1 and a given positive integer n.

a) Write pseudocode to solve this problem. Make sure to clearly define the variables and provide step-by-step instructions for your algorithm.
b) After writing the pseudocode, analyze the time complexity of each line in terms of the input size n. Provide a brief explanation for each line's time complexity analysis.
c) Finally, calculate and provide the overall time complexity of your algorithm using big-O notation.

(20 points) Question 3

For each group of functions, sort the functions in increasing order of asymptotic (big-O) complexity, and explain your answer.

Note: "increasing order" refers to arranging the functions from the one that grows at the slowest rate to the one that grows at the fastest rate. In other words, it means placing the functions in ascending order based on how quickly their time complexity increases as the input size grows.

Group 1:

$f_1(n) = O(2^n)$

$f_2(n) = O(n!)$

$f_3(n) = O(n^3)$

$f_4(n) = O(n^2)$

$f_5(n) = O(n \log n)$

Group 2:

$$
\begin{aligned}
f_1(n) &= 2^{2^{1000000}} \\
f_2(n) &= 2^{100000n} \\
f_3(n) &= \binom{n}{2} \\
f_4(n) &= n\sqrt{n}
\end{aligned}
$$

Search “n choose 2” if you are not familiar with this mathematical expression

(25 points) Question 4:

a) Prove that $f(n) = 3n^2 + 2n + 1$ is in $O(n^2)$. Show the constants c and n_0 that satisfy the definition of Big O.
b) Prove that $f(n) = n^3 + 4n^2 + 2n$ is in $\Theta(n^3)$. Show the constants c1, c2, and n_0 that satisfy the definition of Theta notation.
c) Prove that $f(n) = 2n^2 + 7n$ is in $\Omega(n^2)$. Show the constants c and n0 that satisfy the definition of Omega notation.
d) Given two functions $g(n) = n^2$ and $h(n) = n^3$, determine whether $g(n) = O(h(n))$. Provide a proof for your answer.
e) For the function $f(n) = 5n^2 + 3n \log n + 10$, determine the tightest possible bound using Big O, Omega, and Theta notations. Explain your reasoning.

Remember, proofs for asymptotic notations often involve finding appropriate constants (c and n_0) that satisfy the definitions of Big O, Theta, and Omega. Provide clear explanations and reasoning in your answers.