CSE 321 - Homework 1

Due date: 31/10/2022, 16:45

1. **20pts.** Sort the following functions in increasing order of asymptotic growth. Prove that $T_i(n) \in O(T_j(n))$ for every $T_i < T_j$ where $1 \leq i, j \leq 8$ by using limit approach.

- $T_1(n) = 3logn + 3$
- $T_2(n) = 4log(logn)$
- $T_3(n) = n^5 + 8n^4$
- $T_4(n) = 2000n + 1$
- $T_5(n) = (\frac{n}{6})^2$
- $T_6(n) = 3^n + n^2$
- $T_7(n) = n^n + 1000n$
- $T_8(n) = 2^n + n^3$

2. **20pts.** For the following functions prove whether $f(n) \in O(g(n))$, $f(n) \in \Omega(g(n))$ or $f(n) \in \theta(g(n))$ by using limit approach.

(a) $f(n) = 99n$ and $g(n) = n$

(b) $f(n) = 2n^4 + n^2$ and $g(n) = (logn)^6$

(c) $f(n) = \sum_{x=1}^{n} x$ and $g(n) = 4n + logn$

(d) $f(n) = 3^n$ and $g(n) = 5\sqrt{n}$

3. **20 pts** Examine the following algorithm and answer the questions.

```
int myFunction (int nums[], int n)
{
    for (int i = 0; i < n; i++){
      int count = 1;
      for (int j = i + 1; j < n; j++)
        if (nums[j] == nums[i])
          count++;
      if (count > n / 2)
        return nums[i];
    }
return -1;
}
```

(a) What does the algorithm do? Explain input/output variables.

(b) What is the time complexity of the algorithm? Analyze the worst and best cases.

4. **20 pts** Examine the following algorithm and answer the questions.

```
int myFunction2 (int nums[], int n)
{
    int i, *map, max = 0;

    for (i = 0; i < n; i++)
        if (nums[i] > max)
            max = nums[i];

map = (int *) calloc (max + 1, sizeof (int));

    for (i = 0; i < n; i++)
        map[nums[i]]++;

    for (i = 0; i < n; i++)
        if (map[nums[i]] > n / 2)
            return nums[i];
return -1;
}
```

(a) What does the algorithm do? Explain input/output variables.

(b) What is the time complexity of the algorithm? Analyze the worst and best cases.

5. **10pts.** Compare the algorithms in Question 3 and 4 in terms of time complexity and space used. Explain what makes them better.

6. **10pts.** Consider you are given 2 arrays as follows: $A = [a_1, a_2, \ldots, a_n]$ and $B = [b_1, b_2, \ldots, b_m]$ Describe an algorithm for each of the following problems. Write the pseudo-code of the algorithm and analyze the time complexity of the worst and best cases.

(a) Finding $max\{a_i * b_j\}$ where $1 \leq i \leq n$ and $1 \leq j \leq m$.

(b) Sorting all elements of A and B in descending order as a single array, e.g. $\{a_3, b_1, b_7, \ldots\}$

(c) Adding an element to one of the arrays.

(d) Deleting an element from one of the arrays.