Tutorial 2 TK3043 (Dr Fari)

Please submit your handwritten answer before 31st March 2025 for mark's eligibility.

Section A: Calculate the time and space complexity of these algo:

	The algorithm	*T(n)*	*S(n)*

I	`def first_trace(items):`
	`  print items[0]`
	
	`  mid = len(items) / 3`
	`  index = 0`
	`  while index < mid:`
	`  print items[index]`
	`  index += 1`
	`  for time in xrange(500):`
	`  print "hi"`		

2	`ALGORITHM forth_trace(A[], B[])`
	`//Output: C = AB`
	`for i <- 0 to n-1 do`
	`  for j <- 0 to n-1 do`
	`  C[i, j] = 0.0`
	`  for k <- 0 to n-1 do`
	`  C[i, j] = C[i, j] + A[i, k]×B[k, j]`
	`return C`		

3	`Algorithm mysterytoo(S[n*n])`
	`//input a matrix of n*n`
	`i<-0`
	`j<-1`
	`While i<n`
	`Return i<-i+n`		

Section B: Section B: Cases and recursive

1. Please find from the internet, what is the worse case scenario for the algorithms below. Make sure to fill in the reason as well. You will need to present the reason during tutorial. Tips: get the pseudocode handy for explanations

Algorithm	Best	Average	Worst
Sequential search			
linear sort			
binary search			

**note that for binary search, it is assumed that the array is already sorted

2. Recursive functions

i. Show the analysis of the time complexity using the recursive tree for each algorithm (2 marks)

ii. Derive the recurrence relation of the algorithm. (1 marks)

iii. Determine the time complexity with substitution method (2 marks)

```
function fact(n)
    if n == 0 then
        return 1
    else
        return n * fact(n - 1)
```

```
function binarch(arr, low, high, key)
    if low > high then
        return -1
    mid = (low + high) / 2
    if arr[mid] == key then
        return mid
    else if arr[mid] > key then
        return binarch(arr, low, mid-1, key)
    else
        return binarch(arr, mid+1, high, key)
```

Section C: Asymptotic notation and theorem proving

i. Prove that the **upper bound for** running time $T(n) = n^3 + 20n + 1 \ is \ O(n^3)$

ii. Prove that the **lower bound** below are correct for all n>1
   a. $T(n) = n^3 + 20n \ is \ \Omega(n^2)$
   b. $T(n)=3n^2+4n+2, \ g(n)=n^2$

iii. Determine if f(n)=O(g(n)), g(n)=O(f(n)), or both (**average bound**). Tips: refer to the order of growth sequence
   i. $f(n) = 8n + 20, \ g(n) = \frac{n^2}{2}$
   ii. $f(n) = \frac{n^3-5}{2}, \quad g(n) = 7n$