Lab Week1 Day3

Q 1.Beautiful algorithm (A, n)

Sum 🡨 0

for I 🡨 0 to n-1 do

sum 🡨 sum + a[i]

return sum

A beautiful algorithm is an algorithm with the same best and worst time complexity, which is o(n).

Q2. The order of the time complexity of 2^n, 2^(2n), 2^(n + 1), 2^( 2^n) is

2^n, 2^ (n + 1), 2^(2n), 2^(2^n)

Q3. Mention one algorithm you know for each of the time complexities listed.

O(1) push and pop operations in an array

O(n) search a number in doubly linked list

O(log n) Binary search

O(n log n) Quick sort

O(n^2) Bubble sort

O(n^3) matrix multiplication

O(2^n) Fibonacci number

Q4. Apply Master Theorem and determine the time complexity of

1. Fibonacci number

We can’t apply master theorem for Fibonacci numbers because the value of b is not > 1.

1. Binary search

T(n) = T(n/2) +c

a= 1

b=2

c= cn^k

k=0

a = b ^k

1= 1

So the time complexity is s Θ (n^k logn n)

= Θ(log n )

Q5. Practice Master theorem. It is a very important result in Analysis of algorithms. There are many resources on the internet. Show three different examples covering three possible cases. Show your detailed work.

1. T(n)= 2 T(n/4) + 6n^2

a = 2 , b = 4 k = 2

2 < 4^2

Θ( n^k )

2 T(n) = 3 T(n/2) + 50n

a= 3 , b = 2 k = 0

3 > 2^0

Θ(n ^ logb a )

3. Let T(n) = T( n/ 2) + 1 2 n

a = 1, b = 2 k = 0

1 = 2^0

Θ( n^k log n ) = (log n)