Homework 1

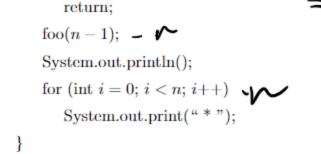
Thursday, September 9, 2021

3:01 PM

1. For each of the following pairs of functions, indicate whether it is one of the three cases: $f(n) = O(g(n)), f(n) = \Omega(g(n)), \text{ or } f(n) = \Theta(g(n)).$ For each pair, you only need to give your answer and the proof is not required. (15 points)

- (a) $f(n) = n^2 + 35n + 6$ and $g(n) = n^4 + 12$. \rightarrow $\{(n) = 0 \text{ (9(N))}\}$
- (b) $f(n) = 5 + \log^2 n + 15n \text{ and } g(n) = 3\log^8 n + \log^3 n. 5$ f(n) = $\Omega(3^n)$
- (c) $f(n) = 64n^4 + 4p^2$ and $g(n) = x^4 + 3n^8 + 10n^2$. \rightarrow f(n): \bigcirc (g(n))
- (d) $f(n) = n \log n$ and $g(n) = y + n^2$. -7 f(N) = o(g(n))
- (e) $f(n) = 2^n$ and $g(n) = 4n^{1000} + 7n^{1000} 6$. $f(n) = \sqrt{2} \left(g(n)\right)$
- For each of the following program fragments, give the running time using big-O notation of n. You only need to give the answer. (15 points)
 - (a) sum = 0; 1for (i = 0; i < n; i++) sum + +; -n
 - (b) sum = 0;for (i = 0; i < n; i++)for (k = 0; k < n * n; k++) $\Rightarrow)$
 - (c) sum = 0;for (i = 0; i < n; i++)for (k = 0; k < i; k++) \Rightarrow $o(n^2)$
 - (d) sum = 0;for (i = 0; i < n; i++)for (k = 0; k < i * i; k++)sum ++:

3. Give the time complexity of the following recursive function using big-O notation of n. You only need to give the answer. (5 points) void foo(int n) if (n == 1)=> O(V,) System.out.print(" * "); - \(\mu\)



- 4. Consider sorting n numbers in an array Λ[0,...,n-1] by first finding the smallest element of Λ and exchanging it with the element in Λ[0]. Then, find the second smallest element of Λ, and exchange it with Λ[1]. Continue this manner for n-1 iterations. This algorithm is known as selection sort.
 (15 points)
 - (a) Write the pseudocode for this algorithm.

Selection Sort (Array) 3

N= Array (Un

for (i=0; i < n-1; i+t)

min= i

for (j=i+1; j < n; j+t)

if Array [j] < Array [min]

min = j

busser = Array [min]
Array [min] . Array [i]
Array [i] = buster

(b) Give the (worst-case) running time of this algorithm using the big-O notation.

((N2)

(c) Give the **best-case** running time of this algorithm using the big-O notation. $O\left(n^{2-1}\right)$

5. Write a Java program to implement the selection sort algorithm introduced in the last question.

(15 points)

see attached file

6. This exercise is to implement the binary search algorithm discussed in class. (15 points)

See Affoched Sile

7. This exercise is to convince you that exponential time algorithms should be avoided.

(10 points)

Suppose we have an algorithm Λ whose running time is $O(2^n)$. For simplicity, we assume the algorithm Λ needs 2^n instructions to finish, for any input size of n (e.g., if n = 5, Λ will finish after $2^5 = 32$ instructions).

According to Wikipedia, as of April 2021, the fastest supercomputer in the world is the Japanese Fugaku (located in in Kobe, Japan) and can perform about 4.0×10^{17} instructions per second.

Suppose we run the algorithm Λ on Fugaku. Answer the following questions.

(a) For the input size n = 100 (which is a relative small input size), how much time does Fugaku need to finish the algorithm? Give the time in terms of centuries (you only need to give an approximate answer).

(b) For the input size n = 1000, how much time does Fugaku need to finish the algorithm? Give the time in terms of centuries (you only need to give an approximate answer).