Nolan
Anderson
09. 16. 2020
Test 1
CS-317

ALGORITHM Comput V(n) // input : $n \geq 1$

$V \leftarrow 0$
for $i \leftarrow 1$ to n do
$V \leftarrow V + i \times i$
return V

$\boxed{1}$

a) What does this algorithm compute?

$V = V + i^2$

$V = 0 + 1^2 = 1$

$V = 1 + 2^2 = 5$

$V = 5 + 3^2 = 14$

The sum of the squares from $1 \to n$

IF n is 50

$n = 3$

b) What is its basic operation?

While there is addition, its basic operation is $i \times i$, or n^2. So mutiplication is the basic operation.

c) how many times is the operation computed?

n times. It only goes n times because there is only one operation inside one loop from $1 \to n$

d) Give an alternate solution that can execute in $O(1)$

$G(n) = G(n-1) + n^2$

$G(1) = G(0) + 1^2 = 1$

$G(2) = G(1) + 2^2 = 5$

$\frac{n(n-1)}{2}$

2)

		Work	Answer
a)	$\log_2 n$	$\log_2 4n - \log_2 n =$	2 times
b)	n^2	$(4n^2)/n^2 =$	4^2 times
c)	$\sqrt{4n}$	$\sqrt{16n}/\sqrt{4n} =$	2 times

3)

a) Alg (n)

// computes 2^n recursively w/: $2^n = 2^{n-1} + 2^{n-1}$

if $n = 0$, return 1

else return $Alg(n-1) + Alg(n-1)$

→ put both b and C here.

b/c) $Alg(n) = 2A(n-1) + 1$ → b

$= 2[2A(n-2)+1]+1 = 2^2 A(n-2) + 2 + 1$

$= 2^2[2A(n-3)+1] + 2 + 1 = 2^3 A(n-3) + 2^2 + 2 + 1$

$= 2^i A(n-i) + 2^{i-1} + 2^{i-2} + \ldots + 1$

$= 2^n Alg(0) + 2^{n-1} + 2^{n-2} + \ldots + 1$

$= 2^{n-1} + 2^{n-2} + \ldots + 1$

$\boxed{= 2^{n-1} + 1}$ → c

d) For i → n do ?

$n = 2 \cdot i$

This might be better since it is only performing one operation per iteration.