Name: Liam Brew Date: 02.21.2020

Pledge: I pledge my honor that I have abided by the Stevens Honor System – Liam Brew

Page 67

1. **ALGORITHM** *Mystery*(n)

```
//Input: A nonnegative integer n
S ← 0
for i ← 1 to n do
    S ← S + i * i
return S
```

a. This algorithm computes the sum of squares of the first n positive integers.
b. The basic operation of this algorithm is the for loop that iterates from 1 to n-1.
c. The basic operation of this algorithm is executed n times.
d. This algorithm has an efficiency class of $\theta(n)$.
e. A more efficient implementation of this algorithm is achieved using the summation for the sum of squares of the first n positive integers, whose formula is $\frac{n*(n+1)*(2n+1)}{6}$. This new algorithm has an improved efficiency class of $\theta(1)$.

Page 76

Question 1: Recurrence Relations

a. $x(n) = x(n-1) + 5 \; for \; n > 1, x(1) = 0$

Step 1: $x(n-1) = x(n-1-1) + 5$

$x(n) = x(n-1) + 10$

Step 2: $x(n-2) = x(n-1-2) + 5$

$x(n) = x(n-3) + 15$

Step 3: $x(n) = x(n-i) + 5 * i$

Step 4: Initial condition of x(1) = 0 → n – i = 0 → i = n – 1

Step 5: $x(n) = x(n-(n-1)) + 5(n-1)$

$x(n) = x(1) + 5(n-1)$

i. $x(n) = 5(n-1)$

b. $x(n) = 3x(n-1) \; for \; n > 1, x(1) = 4$

Step 1: $x(n-1) = 3x(n-1-1)$

$x(n) = 9x(n-2)$

Step 2: $x(n-2) = 3x(n-2-1)$

$x(n) = 27x(n-3)$

Step 3: $x(n) = 3^i x(n-i)$

Step 4: Initial condition of x(1) = 4 → n – i = 1 → i = n – 1

Step 5: $x(n) = 3^{n-1}x\big(n-(n-1)\big)$

$x(n) = 3^{n-1}x(1)$

$x(n) = 3^{n-1} * 4$

i. $x(n) = 4 * 3^{n-1}$

c. $x(n) = x(n-1) + n \; for \; n > 0, x(0) = 0$

Step 1: $x(n-1) = x(n-1-1) + (n-1)$
$x(n) = x(n-2) + (n-1) + n$
Step 2: $x(n-2) = x(n-1-2) + (n-2)$
$x(n) = x(n-3) + (n-2) + (n-1) + n$
Step 3: $x(n) = x(n-i) + (n-i+1) + \cdots + (n-1) + n$
Step 4: Initial condition of x(0) = 0 → n – i = 0 → n = i
Step 5: $x(n) = x(n-n) + (n-n+1) + \cdots + (n-1) + n$
$x(n) = 0 + 1 + \cdots + (n-1) + n$
i. $x(n) = \frac{n*(n+1)}{2}$

d. $x(n) = x\left(\frac{n}{2}\right) + n \; for \; n > 1, x(1) = 1 \; \left(solve \; for \; n = 2^k\right)$
Step 0: $x(2^k) = x(2^{k-1}) + 2^k$
Step 1: $x(2^{k-1}) = x(2^{k-2}) + 2^{k-1}$
$x(2^k) = x(2^{k-2}) + 2^{k-1} + 2^k$
Step 2: $x(2^{k-2}) = x(2^{k-3}) + 2^{k-2}$
$x(2^k) = x(2^{k-3}) + 2^{k-2} + 2^{k-1} + 2^k$
Step 3: $x(2^k) = x(2^{k-i}) + 2^{k-i+1} + 2^{k-i+2} + \cdots + 2^k$
Step 4: Initial condition of x(1) = 1 → $2^{k-i} = 1$→ $2^{k-i} = 2^0$→ k – i = 0 → i = k
Step 5: $x(2^k) = x(2^{k-k}) + 2^{k-k+1} + 2^{k-k+2} + \cdots + 2^k$
$x(2^k) = x(2^0) + 2^1 + 2^2 + \cdots + 2^k$
$x(2^k) = 1 + 2^1 + 2^2 + \cdots + 2^k$
$x(2^k) = \sum_{i=0}^{N} 2^i = \frac{2^{k+i}-1}{1}$
$x(2^k) = 2 * 2^k - 1$
i. $x(n) = 2n - 1$

e. $x(n) = x\left(\frac{n}{3}\right) + 1 \; for \; n > 1, x(1) = 1 \; \left(solve \; for \; n = 3^k\right)$
Step 0: $x(3^k) = x(3^{k-1}) + 1$
Step 1: $x(3^{k-1}) = x(3^{k-1-1}) + 1$
$x(3^k) = x(3^{k-2}) + 1 + 1$
$x(3^k) = x(3^{k-2}) + 2$
Step 2: $x(3^{k-2}) = x(3^{k-1-2}) + 1$
$x(3^k) = x(3^{k-3}) + 1 + 2$
$x(3^k) = x(3^{k-3}) + 3$
Step 3: $x(3^k) = x(3^{k-i}) + i$
Step 4: Initial condition of x(1) = 1 → $3^{k-i} = 1$→ $3^{k-i} = 3^0$→ k – i = 0 → i = k
Step 5: $x(3^k) = x(3^{k-k}) + k$
$x(3^k) = x(3^0) + k$
$x(3^k) = 1 + k$
$\mathrm{Log}_3 \, n = k$
i. $x(n) = 1 + \mathrm{Log}_3 \, n$

Question 3: Algorithm analysis

a. $S(n) = S(n-1) + 1 \; for \; n > 1, x(1) = 1$

i.
Step 1: $S(n-1) = S(n-1-1) + 1$
$S(n) = S(n-2) + 1$
$S(n) = S(n-2) + 2$
Step 2: $S(n-2) = S(n-1-2) + 1$
$S(n) = S(n-3) + 2 + 1$
$S(n) = S(n-2) + 3$
Step 3: $S(n) = S(n-i) + i$
Step 4: Initial condition of x(1) = 1 → n – i = 1 → i = n -1
Step 5: $S(n) = S(n-(n-1)) + n - 1$
$S(n) = S(1) + n - 1$
$S(n) = n, \theta(n)$

b. The non-recursive algorithm, using the summation formula of $\sum_{i=1}^{n} i^3 = \frac{n^2(n+1)^2}{4}$, can be achieved in a constant time of $\theta(1)$.