Homework Assignment 2

Katie Prescott
2/23/16

I pledge my honor that I have abided by the Stev[ens] honor System. [signature]

P67: 4 (a, b, c, d, e), P76: 1 (a, b, c, d, e), P76-77: 3 (a, b)

Pg 67:

4. a Mystery(n)

```
Mystery(n)
// input: Non-neg. #
S ← 0
for i ← 1 to n do
    S ← S + i * i
return S
```

a. What does the algorithm compute?

The algorithm computes the sum of the squares of i from 1 to n.

$$\boxed{\sum_{i=1}^{n} i * i}$$

b. What is it's basic operation?

$\boxed{\text{multiplication}}$

c. How many times is the basic operation executed?

$\boxed{\text{n times.}}$

d. What is the efficiency class of this algorithm?

$\boxed{\Theta(n)}$

e. Suggest an improvement; if no improvements can be made, prove it. What is new efficiency class?

```
Mystery(n)
// input: non-neg. #

S = (n(n+1)(2n+1))/6;
return S.
```

$\boxed{\Theta(1)}$

76: 1. Solve the following recurrence relations

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

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

$x(n) = x(n-2) + 5 + 5 = x(n-2) + 10$

$x(n-2) = x(n-3) + 5$

$x(n) = x(n-3) + 10 + 5 = x(n-3) + 15$

$\boxed{x(n) = x(n-k) + 5k}$ $x(1) = 0$

$n - k = 1$

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

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

$\boxed{x(n) = 5n - 5}$

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

$x(n-1) = 3 \cdot x(n-2)$

$x(n) = 3 \cdot 3 \cdot x(n-2)$

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

$x(n-2) = 3 \cdot x(n-3)$

$x(n) = 9 \cdot 3 \cdot x(n-3) = 27\, x(n-3)$

$x(n) = 3^k x(n-k)$ $x(1) = 4$

$n - k = 1$

$k = n - 1$

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

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

$\boxed{x(n) = 4 \cdot 3^{n-1}}$

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

$x(n-1) = x(n-2) + n - 1$

$x(n) = x(n-2) + n + n - 1 = 2n - 1$

$x(n-2) = x(n-3) + n - 2$

$x(n) = x(n-3) + n - 2 + n - 1 + n = 3n - 3$

$x(n-3) = x(n-4) + n - 3$

$x(n) = x(n-4) + n - 3 + n - 2 + n - 1 + n = 4n - 6$

$x(n-4) = x(n-5) + n - 4$

$x(n) = x(n-5) + n - 4 + n - 3 + n - 2 + n - 1 + n = 5n - 10$

$x(n) = x(n-k) + (n-k-1) + \dots + (n-1) + n$

$x(n) = x(0) + 1 + 2 + \dots + n$

$\boxed{x(n) = \frac{n(n+1)}{2}}$

d. $x(n) = x(\frac{n}{2}) + n \quad n > 1 \quad x(1) = 1$ (solve for 2^k)

$x(\frac{n}{2}) = x(\frac{n}{4}) + \frac{n}{2}$

$x(n) = x(\frac{n}{4}) + \frac{n}{2} + n = x(\frac{n}{4}) + \frac{3n}{2}$

$x(\frac{n}{4}) = x(\frac{n}{8}) + \frac{n}{4}$

$x(n) = x(\frac{n}{8}) + \frac{n}{4} + \frac{n}{2} + n = x(\frac{n}{8}) + \frac{7n}{4}$

$x(\frac{n}{8}) = x(\frac{n}{16}) + \frac{n}{8}$

$x(n) = x(\frac{n}{16}) + \frac{n}{8} + \frac{n}{4} + \frac{n}{2} + n = x(\frac{n}{16}) + \frac{15n}{8}$

$x(n) = x\left(\frac{n}{2^k}\right) + \frac{2^k - 1}{2^{k-1}} n$

$x(n) = x\left(\frac{2^k}{2^k}\right) + \frac{2^{\lg n} - 1}{2^{\lg n - 1}} n \qquad x(1) = 1$

$\frac{n}{2^k} = 1, \quad n = 2^k$

$k = \lg n$

$= x(1) + \frac{n-1}{n/2} n$

$x(n) = 1 + 2(n-1)$

$\boxed{x(n) = 2n - 1}$

e. $x(n) = x(\frac{n}{3}) + 1 \quad n > 1, \ x(1) = 1$ (solve for 3^k)

$x(\frac{n}{3}) = x(\frac{n}{9}) + 1$

$x(n) = x(\frac{n}{9}) + 2$

$x(\frac{n}{9}) = x(\frac{n}{27}) + 1$

$x(n) = x(\frac{n}{27}) + 3$

$x(1) = 1$

$\frac{n}{3^k} = 1 \qquad n = 3^k$

$k = \log_3 n$

$x(n) = x\left(\frac{n}{3^k}\right) + k$

$x(n) = x\left(\frac{3^k}{3^k}\right) + \log_3 n$

$x(n) = x(1) + \log_3 n$

$\boxed{x(n) = 1 + \log_3 n}$

76-77: 3. Consider the following recursive algorithm for computing the sum of the first n cubes $S(n) = 1^3 + 2^3 + 3^3 + \dots + n^3$

S(n)
// input: pos. int. n
// output: sum of first n cubes

if n=1 return 1
else return S(n-1) + n*n*n

a. Set up & solve a recurrence relation for the # of times the algorithms basic operation is executed

$M(n) = M(n-1) + 2 \quad , M(1) = 0$

$M(n-1) = M(n-2) + 2$

$M(n) = M(n-2) + 2 + 2 = M(n-2) + 4$

$M(n-2) = M(n-3) + 2$

$M(n) = M(n-3) + 2 + 4 = M(n-3) + 6$

$M(n-3) = M(n-4) + 2$

$M(n) = M(n-4) + 2 + 6 = M(n-4) + 8$

$M(n) = M(n-k) + 2k$

$M(1) = 0$

$n-k = 1$

$k = n-1$

$M(n) = M(n-(n-1)) + 2(n-1)$

$M(n) = M(1) + 2(n-1)$

$M(n) = 0 + 2(n-1)$

$\boxed{M(n) = 2n - 2}$

b. How does this algorithm compare with the straightforward, nonrecursive algorithm for this sum?

- The performance / complexity would be the same, the only difference would be the amount of Space in Memory that is used would be less for the non-recursive algorithm.