CS302 Data Structures

Spring 2010 – Dr. George Bebis

Homework 2 - Solutions

1.

1000   10lgn 2^{lgn} 100n n^2 $4n^2$ 2^n

(Slowest → Fastest)

2.

$n^2 \geq 2^{n/4}$ for $n \leq 8$

So, $2^{n/4}$ becomes larger for $n \geq 9$

3.

(a) sum = 0; $O(1)$

$O(2n)$ for(i=1; i<=2*n; i++)

$O(1)$ → sum = sum + 1;

Total: $O(2n) = O(n)$

(b) sum = 0; $O(1)$

$O(n^2)$ for(i=1; i<=n*n; i++)

$O(1)$ → sum = sum + 1;

Total: $O(n^2)$

(c) sum=0; $O(1)$

$O(n)$ for(i=1; i<=n; i++)

$O(1)$ → sum = sum + n;

Total: $O(n)$

(d) sum = 0; $O(1)$

$O(n)$ for(i=1; i<=n i++)

$O(i)$ for(j=1; j<=i; j++)

$O(1)$ sum = sum + i;

where $i = 1, 2 \ldots, n$

when $i=1$: $j \leq 1$ — 1 time

$i=2$: $j \leq 2$ — 2 times

$\vdots$

$i=n$: $j \leq n$ — n times

Total $1 + 2 + \cdots + n = \frac{n(n+1)}{2} = O(n^2)$

(e) sum = 0; $O(1)$

$O(1)$ for(i=1; i<=100; i++)

$O(n)$ for(j=1; j<=n; j++)

$O(1)$ sum = sum + i;

Total: $O(n)$

(f) sum = 0; $O(1)$

for(i=1; i<=n; i++) $O(n)$

for(j=1; ~~i~~j<=n; j*=2) $O(\lg n)$

sum = sum + 1; $O(1)$

Total: $O(n \lg n)$

4.

a. $5n+n^2 -2$ → $O(n^2)$
b. 7 → O(1)
c. 4n+10lgn+25 → O(n)
d. 3 + 4 lgn → O(lgn)
e. $n^2 + n^3 + 10$ → $O(n^3)$

5(a)

See my notes (i.e., slide #25 from the lecture on “Comparison of Algorithms)

5(b)

The overall complexity will be determined by the “slowest” function; that is, it will be O(n)