Albert Owusu Asare

Algorithms homework

February 2 2016

7, 8a-d, 13, 18, 23, 32, 1 interview

Question 2.1

We shall use summations to find what the algorithm evaluates

we have:

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \sum_{k=1}^{j} 1 = \sum$$

$$= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} j$$

$$= \sum_{i=1}^{n-1} \left(\sum_{j=1}^{n} - \sum_{j=1}^{i} \right)$$

$$= \sum_{i=1}^{n-1} \frac{n(n+1)}{2} - \left[\frac{i(i+1)}{2} \right]$$

$$= \frac{1}{2} n(n+1)(n-1) - \frac{1}{2} \sum_{i=1}^{n-1} i^2 + i$$

we use wolphram alpha to solve the summation

$$= \frac{1}{2} n(n+1)(n-1) - \frac{1}{2} \left(\frac{1}{3} n(n+1)(n-1) \right)$$

$$= \cancel{\frac{1}{2}} n(n+1)(n-1) \left[\frac{1}{2} - \frac{1}{6} \right]$$

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

Big oh worst $O(n^3)$

Question 2-8

(1) $f(n) = \log n^2$; $g(n) = \log n + 5$

Notice that $\log n^2 = 2 \log n$

Notice also

$$2 \log n \leq 2 (\log n + 5)$$

hence there exist some $c_1 = 2$ such that $f(n) \leq c_1 \cdot g(n)$

Notice also that

$$2 \log n \geq 0 \cdot (\log n + 5)$$

$$2 \log n \geq 0$$

hence there exist some $c_2 = 0$ such that $f(n) \geq c_2 \cdot g(n)$

Now because there are constants c_1 & c_2 such that $f(n) \leq c_1 \cdot g(n)$ $f(n) \geq c_2 \cdot g(n)$. This means that $f(n) = \Theta(g(n))$

(2) $f(n) = \sqrt{n}$; $g(n) = \log(n^2)$

From section 2.9.2, we say $f(n)$ dominates $g(n)$ if $\lim_{n \to \infty} g(n)/f(n) = 0$

Notice $\lim_{n \to \infty} \frac{g(n)}{f(n)} = \frac{2 \log n}{\sqrt{n}} = 0$

As n gets infinitely large $\log n$ ~~dominates $\sqrt{n}$~~ gets smaller ~~large~~ than $g(n)$ so we approach ~~infinity~~ 0 hence $f(n)$ dominates

$$\boxed{f(n) = \Omega(g(n))}$$

(3) $f(n) = \log^2(n)$; $g(n) = \log(n)$

$\lim_{n \to \infty} \frac{g(n)}{f(n)} = \frac{\log(n)}{\log^2(n)} = 0$

as n gets infinitely large $\log^2(n)$ dominates $\log n$ so we approach 0. This means $f(n)$ dominates $g(n)$

$$\boxed{f(n) = \Omega(g(n))}$$

(7) $f(n) = n\log(n) + n$; $g(n) = \log(n)$

$$\lim_{n \to \infty} \frac{g(n)}{f(n)} = \frac{\log(n)}{n\log(n) + n} = 0$$

As n gets infinitely long the denominator gets bigger than the numerator so f(n) dominates

$$f(n) = \Omega(g(n))$$

2.13 $f_1(n) = O(g_1(n))$ implies there exist some c_1 such that $f_1(n) \le c_1 g_1(n)$

$f_2(n) = O(g_2(n))$ implies there exist some c_2 such that $f_2(n) \le c_2 g_2(n)$.

Now $f_1(n) + f_2(n) \le c_1 g_1(n) + c_2 g_2(n)$

$\le \max(c_1, c_2)\, g_1(n) + \max(c_1, c_2)\, g_2(n)$

$\le \max(c_1, c_2)\,[g_1(n) + g_2(n)]$

let $\max(c_1, c_2) = c_3$

$= c_3\,[g_1(n) + g_2(n)]$

By definition we have shown that $f_1(n) + f_2(n) = O(g_1(n) + g_2(n))$

2.18

$\lg n$

$\sqrt{n}$

$\lg \lg n$

$\ln x$

n

$n \lg n$

$(\lg n)^2$

$n^{1+\varepsilon}$ $0 < \varepsilon < 1$

n^2

$n^2 + \log n$

n^3

$7n^5 - n^3 + n$

e^n

2^{n-1}

2^n

$n!$

2.23

(a) Yes depending on the input, an algorithm with $O(n^2)$ worst-case may do well ($O(n)$). Ex insertion sort may be $O(n)$ if the input array is mostly sorted. This is just an upperbound of how bad your algorithm could be.

(b) Yes, similar reasoning the algorithm could yet be $O(n)$ on all inputs. Worst case just gives ~~a~~ us a bound to the worst possible the algorithm could perform.

(c) Yes, average means on some inputs the algorithm could do really well. Thus better than n^2 and in ~~that~~ others very low, but averages/means out to n^2.

d) No average means some inputs will do letter and worse the $O(n^2)$

(e) Yes, the leading term is what matters. In both cases we realise an average case $\Theta(n^2)$.

2.32

We are to show that
$$P(k) = 1^2 + 2^2 + 3^2 - 4^2 + \dots + (-1)^{k-1} k^2 = (-1)^{k-1} k(k+1)/2 \quad \forall k \in \mathbb{Z},\ k \geq 1$$

e shall use induction on $\mathbb{Z}$.

Base case: when $k = 1$
$$1^2 = (-1)^0 \frac{1(1+1)}{2} = 1 \quad \text{hence proof.}$$

Inductive case: We shall assume $P(k)$ holds and show that $P(k+1)$ holds.

Thus we show that
$$P(k+1) = 1^2 - 2^2 + 3^2 - 4^2 + \dots + (-1)^{k-1} k^2 + (-1)^k (k+1)^2 = (-1)^k (k+1) \frac{(k+2)}{2}$$

$$^2 - 2^2 + 3^2 - 4^2 + \dots + (-1)^{k-1} k^2 + (-1)^k (k+1)^2 = P(k) + (-1)^k (k+1)^2$$
$$= \frac{(-1)^{k-1} k(k+1)}{2} + (-1)^k (k+1)^2 \quad \text{inductive hypothesis}$$
$$= (-1)^k (k+1) \left[-\frac{k}{2} + (k+1) \right]$$
$$= (-1)^k (k+1) \left[\frac{k+2}{2} \right] \quad \square$$

have been able to show that $\forall k \in \mathbb{Z}$, $k \geq 1$ our claim holds