Stamati Morellas
CS311 Homework 1
Submitted: 9/13/19

Stamati Morellas
Section 6 - Tuesday 1:10
9/11/2019

1a. Let $c=89$, and $n \geq 1$

For every $n \geq 1$:

$$8n^2+35n+46 \leq 8n^2+35n^2+46n^2 = 89n^2$$

Therefore, $8n^2+35n+46 \in O(n^2)$ is $\boxed{\text{True}}$

1b. $2^{2^{n+1}} \in O(2^{2^n})$?

Let $f(n) = 2^{n+1}$

Let $g(n) = 2^n$

For every $n \geq 1$:

$$f(n) \leq cg(n) = 2^{n+1} \leq 2(2^n) \Rightarrow 2^{n+1} \leq 2^{n+1} \Rightarrow c=2$$

$\therefore$ $2^{2^{n+1}} \in O(2^{2^n})$ is $\boxed{\text{True}}$ for every $n \geq 1$ with constant $c=2$

1c. $n^3(5+\sqrt{n}) \in O(n^3)$?

Let $f(n) = n^3(5+n^{1/2}) = 5n^3+n^{3/2}$

Let $g(n) = n^3$

For every $n \geq 1$:

Since $\sqrt{n}$ will always be ≥ 1 c would need tobe

$c \geq (5+n^{1/2})$ in order for $f \in O(g)$. $(\sqrt{n} \leq c-5)$

$f \notin O(n^3)$ because if it was, then $(5+\sqrt{n}) \leq c \Rightarrow N \leq c-5^2$

This does not happen for every $n \geq 0$.

So if $n=1$ and $c=2$, then $1 \leq -3$ is not true.

$\therefore$ $\boxed{n^3(5+n^{1/2}) \notin O(n^3)}$

Stamati Morellas
Section 6

1d. Prove for all $a \geq 0$, $2^{n+a} \in O(2^n)$

Let $f(n) = 2^{n+a}$ $\quad f(n) \leq cg(n)$ in order to hold true

Let $g(n) = 2^n$

$2^{n+a} \leq c2^n$

if $c = 2^a$, then this holds $\boxed{\text{True}}$ for all $a \geq 0$ since $2^0 = 1$ which satisfies the constraints for c, since it'll always be $c \geq 1$

$\therefore$ This is $\boxed{\text{True}}$

1e. For a positive-valued function h,

$h \cdot f(n) \leq h \cdot c \cdot g(n)$ in order for $f \in O(g)$

Let $f(n) = 2^{n+1}$ Let $h(n) = n+1$

Let $g(n) = 2^n$

$\forall n \geq 1$:

$(n+1)(2^{n+1}) \leq c \cdot (n+1)(2^n)$

When $c = 2$:

$(n+1)(2^{n+1}) \leq 2(n+1)(2^n) \Longrightarrow (n+1)(2^{n+1}) \leq (n+1)(2^{n+1})$ $\boxed{\text{True}}$

$\therefore$ $f \cdot h \in O(gh)$ is $\boxed{\text{True}}$

2a.

Outer loop runs $\underline{n}$ times: $O(n)$ $\quad \sum_{i=1}^{n} c$

Inner loop runs $\underline{n}$ times: $O(n)$ $\quad \sum_{j=i}^{n} c$

Total Runtime: $n \cdot n = n^2$ $\quad \sum_{i=1}^{n} \sum_{j=i}^{n} c_1 + \sum_{i=1}^{n} c_2$

$\therefore$ $\boxed{T(n) = O(n^2)}$

$= n^2 c_1 + n c_2$

Stamati Morellas
Section 6

2b.

Outer loop runs $\underline{n}$ times: $O(n)$ $\sum_{i=1}^{n} c$

Inner loop runs $\underline{n \cdot i}$ times: $O(n^2)$ $\sum_{j=1}^{n \cdot i} c$

Total Runtime: $n \cdot n^2 = n^3$ $\sum_{i=1}^{n}\sum_{j=1}^{n \cdot i} c_1 + \sum_{i=1}^{n} c_2 = c_1 n^3 + c_2 n$

$\therefore \boxed{T(n) = O(n^3)}$

2c. Outer loop runs $\underline{2^n}$ times: $O(2^n)$ $\sum_{i=1}^{2^n} c$

Inner loop runs: $\underline{i}$ times

$i: 1, 2, 4, 8, 16, \ldots, 2^n$

Outer: $\sum_{i=1}^{2^n} (\text{inner}) + \sum_{i=1}^{2^n} c_2$

Inner: $\sum_{j=1}^{n} c_1$

$\Longrightarrow \longrightarrow \sum_{i=1}^{2^n}\sum_{j=1}^{n} c_1 + \sum_{i=1}^{2^n} c_2 = c_2 n + 2c_2 + c_1 n^2 + 2c_1 n$

$\therefore \boxed{T(n) = O(n^2)}$

2d. Outer loop runs $\underline{\log n}$ times: $O(\log n)$

Inner loop runs $\underline{n}$ times: $O(n)$

$\sum_{i}^{\log n} c_2 + \sum_{i}^{\log n}\sum_{j=1}^{n} c_1 = c_2 \log n + c_1 n \log n$

$\therefore \boxed{T(n) = O(n \log n)}$

Stamati Morellas
Section 6

3. This algorithm is $\boxed{\text{incorrect}}$. By using a different algorithm, the runtime for this can be more time-efficient. The Greedy Algorithm takes $O(n\log n)$, which is shorter than the proposed. By sorting the jobs by the order of finish times $(f_1 \leq f_2 \leq f_3 \leq \ldots \leq f_n)$ and checking to see if a set of jobs is compatible, this will be a more efficient way to schedule the jobs.

4. <u>GCD Runtime:</u>

- 9-digit primes: 5717ms Numbers: 674,506,111 and 899,809,343
- 10-digit primes: 12613 ms Nums: 1,500,450,271 and 2,860,486,313

<u>Fast GCD Runtime:</u>

- 9-digit primes: 0ms
- 10-digit primes: 0 ms

5. Function(A, t)

$$A, t \longrightarrow \boxed{\text{Func}} \longrightarrow P_A(t)$$

```
Define sum = 0
For i in range [A.length - 1, 0]
    sum = sum + A[i] · pow(t, i)
Return sum
```