CS 2500

SP 2024

Homework 1

#1

Prob. 1

Implement Karatsuba's algorithm in Python.

Prob 2

Empirical run-time analysis of Karatsuba's algorithm in Python. → Similar to Insertion Sort

Prob. 3

Let $a(n) = n^2 \log_2 n$, $b(n) = 2^n$, $c(n) = 2^{2^n}$, $d(n) = n^{\log_2 n}$

and $e(n) = n^2$.

Claim 3.1 : $e(n) = n^2 = O(2^n) = O[b(n)]$

Proof : If $e(n) = O[b(n)]$, there are two constants (k, n_0) s.t. $e(n) \leq k \cdot b(n) \quad \forall \ n \geq n_0$.

i.e. $n^2 \leq k \cdot 2^n \quad \forall \ n \geq n_0$.

Applying logarithm on both sides,

$$2 \log n \leq \log k + n \cdot \log 2$$

$$\Rightarrow \log n \leq \frac{\log k}{2} + \left(n \log 2 / 2\right)$$

P.T.O.

#2

We know

$$\log n \leq n - 1 \quad \forall \; n \geq 0.$$

Since $\frac{\log 2}{2} \cdot n + \frac{\log k}{2} < n + \frac{\log k}{2}$

We can choose $\frac{\log k}{2} = 0$

$$\Rightarrow \quad k = 1 \quad \text{and} \quad n_0 = 1.$$

$\square$

<u>Claim 3.2</u> : $b(n) = 2^n = O(2^{2^n}) = O[c(n)]$.

Proof : We need to find two constants k, n_0 s.t. $\forall \; n \geq n_0, \quad 2^n \leq k \cdot 2^{2^n}$.

Apply log on both sides, we have

$$n \log 2 \leq \log k + 2^n \cdot \log 2$$

Let $k = 1$, then $n \cancel{\log 2} \leq 2^n \cancel{\log 2}$

(or) $n \leq 2^n$.

Note that $n \leq 2^n \quad \forall \; n \geq 1$.

$\therefore \; k = 1$ and $n_0 = 1$.

$\square$

<u>Claim 3.3</u> : $d(n) = n^{\log n} = O[2^n] = O[b(n)]$.

<u>Proof</u> : Need to find (k, n_0) s.t.

$$n^{\log n} \leq k \cdot 2^n \quad \forall \quad n \geqslant n_0 .$$

Applying logs on both sides,

$$(\log n).(\log n) \leq \log k + n \log 2$$

If $k = 1$, then

$$(\log n)^2 \leq n \log 2 .$$

Multiplying $\frac{1}{4}$ on both sides,

$$\left[\frac{1}{2} \log n\right]^2 \leq \frac{\log 2}{4} . n$$

(or) $$\left[\log(\sqrt{n})\right]^2 \leq \frac{\log 2}{4} . (\sqrt{n})^2$$

We know $\log x \leq x \quad \forall \quad x \geqslant 0$.

$$\therefore (\log \sqrt{n})^2 \leq \frac{\log 2}{4} (\sqrt{n})^2 \quad \forall \quad n \geqslant \text{[redacted]}.$$

$\Rightarrow$ $k = 1$, $n_0 =$ [redacted]. $\square$.

#4

Claim 3.4 : $a(n) = n^2 \log n = O[n^{\log n}] = O[d(n)]$

Proof : Note that $n^2 \log n < n^3$ since $\log n < n$ $\forall n \geq 0$.

Now, $n^3 < n^{\log n}$ for any $\log n > 3$ (or) $n > 2^3 = 8$.

$\therefore$ $n^2 \log n < \underset{\underset{k}{\shortparallel}}{1} \cdot n^{\log n}$ $\forall n > n_0 = 8$

$\square$

Claim 3.5 : $e(n) = n^2 = O[n^2 \log n] = O[a(n)]$.

Proof : Omitted for the sake of brevity as it is a trivial exercise.

Claim 3.6 : Let $f(n) \preccurlyeq g(n)$ means $f(n) = O[g(n)]$

Then, $e(n) \preccurlyeq a(n) \preccurlyeq d(n) \preccurlyeq b(n) \preccurlyeq c(n)$.

Proof : Combine the results in Claims 3.1 to 3.5.

$\square$

Prob. 4

```
Incremental Search (A)
    for i = 1 to A.length
        if A[i] = i
            return TRUE
    return FALSE
```

$T_1(n) = O(n)$

In the worst case, i.e. when A does not contain i s.t. $A[i] = i$.

Now, consider divide-and-conquer style.

```
Binary Search (A)
    mid = A.length / 2

    if mid ≥ 1
        if A[mid] = mid
            return TRUE
        else if A[mid] > mid
            Binary Search (A[1 : mid-1])
        else
            Binary Search (A[mid+1 : n])
    else
        return FALSE.
```

Note that $T_2(n) \le T_2\left(\frac{n}{2}\right) + c$. Since $a = 1 = 2^0 = b^d$,

we have $T_2(n) = O(n^0 \log n) = \underline{O(\log n)}$

Prob. 5

$$T(n) \leq 7 \cdot T\left(\frac{n}{3}\right) + O(n^2)$$

Here, $a = 7$, $b = 3$ and $d = 2$.

Since $a = 7 < 3^2 = b^d$, this recursion falls under Case - 2 of Master method.

$$\Rightarrow T(n) = \underline{\underline{O(n^d) = O(n^2)}}$$

Prob 6 [Extra Credit]

$$T(n) \leq T\left(\lfloor \sqrt{n} \rfloor\right) + 1 \qquad \text{where } T(1) = 1.$$

Note: This is not a standard recursion.

Let $n = 2^{2^k}$. $\Rightarrow 2^k = \log n \Rightarrow k = \log(\log n)$

Then, $T(n) \leq T\left(\lfloor 2^{2^{k-1}} \rfloor\right) + 1$

$$= T\left(2^{2^{k-1}}\right) + 1 \quad \text{--- (1)}$$

Ily, $T\left(2^{2^{k-1}}\right) \leq T\left(2^{2^{k-2}}\right) + 1 \quad \text{--- (2)}$

#7

Substituting 2 in 1, we have

$$T\left(2^{2^k}\right) \leq T\left(2^{2^{k-1}}\right) + 2 \quad — \ (3)$$

Expanding the RHS of 3 iteratively, we get

$$T\left(2^{2^k}\right) \leq T\left(2^{2^{k-i}}\right) + i \quad — \ (4)$$

$$\forall \ i = 1, 2, \ldots, k$$

Finally, after the k^{th} iteration,

$$T(2) \leq T\left(\lfloor \sqrt{2} \rfloor\right) + 1$$

$$= T(1) + 1 = 2. \quad — \ (5)$$

$\therefore$ ~~[illegible]~~ Combining 5 with k^{th} eqn. in 4,

we obtain

$$T(n) \leq T\left(2^{2^0}\right) + k$$

$$= 2 + k = 2 + \log(\log n) = \underline{\underline{O(\log \log n)}}$$