Illinois Institute of Technology
Department of Computer Science

Honesty Pledge

CS 430 Introduction to Algorithms
Spring Semester, 2018

Fill out the information below, sign this sheet, and submit it with the first homework assignment. No homework will be accepted until the signed pledge is submitted.

I promise, *on penalty of failure of CS 430*, not to collaborate with anyone, not to seek or accept any outside help, and not to give any help to others on the homework problems in CS 430.

All work I submit will be mine and mine alone.

I understand that all resources in print or on the web, aside from the text and class notes, used in solving the homework problems *must be explicitly cited.* I understand that failure to cite sources used will result in a score of zero on the problem.

Name (printed)	Signature	Student ID	Date
Christopher Moccom	Christopher S. Moccom	A20385764	17 Jan 18

Morcom

1) Base Case:

$T(2) = 2$

$T(n) = 2T(\frac{n}{2}) + n$ if $n = 2^k$ & $k > 1$

$T(2^k) = 2^k \lg(2^k) = k2^k$

$n = 2^k$

$k = \lg(n)$

Inductive Step:

$$T(2^{k+1}) = 2T(\frac{2^{k+1}}{2}) + 2^{k+1}$$
$$= 2T(2^k) + 2^{k+1}$$

let $T(2^k) = t_k$, hence $T(2^{k+1}) = t_{k+1}$

$$\underbrace{t_{k+1} = 2t_k}_{(E-2)} + \underbrace{2^{k+1}}_{(E-2)} = (E-2)^2$$

series annihilated by

$$t_{k+1} = (\alpha k + \beta)2^k$$
$$= \alpha k 2^k + \beta 2^k$$

$T(2) = 2$, $T(4) = 8$, $T(8) = 24$

hence, $\alpha = 1$, $\beta = 1$

$$t_{k+1} = 2^k(k+1)$$

meaning n must be an exact power of 2.

2) Insertion sort recurrence relation is given by:

$$T(n) = \underbrace{T(n-1)}_{\text{sort } n-1 \text{ elements recursively}} + n \leftarrow \text{insert element into array}$$

$T(n)$ annihilated by $(E-1)^2$ which gives best case of ~~$\Theta(n)$~~ $O(n)$ time.

Worst case time is given by:

$$T(n+2) = T(n+1) + T(n-1) + \Theta(n), \quad T(0)=1, \; n \geq 1$$

which is annihilated by $(E-1)^3$

which gives worst case time complexity of $O(n^2)$

3) In terms of Θ notation, the code has an absolute run time of $O(n+1)$, which is approx $\Theta(n)$ since it is a for-loop of $n-1$ iterations with one line of execution.

4) As n gets extremely large, the order of ascending growth is given by:

$$* \quad \underbrace{2^{2^{n+1}}} < 2^n < n \lg n < n < 2^{\sqrt{2 \lg(n)}} < \underbrace{\lg(\lg^*(n))}$$

grows exponentially small, so it must be the min function approx. given by Ω-notation

$2^{2(n+1)} = \Omega(2^n)$

$n \lg(n) = \Omega(n) \leftarrow$ see problem 1 and example in notes

$$n = \Omega(2^{\sqrt{2 \lg n}}) = \Omega(2^{\sqrt{2} \cdot \sqrt{\lg n}}) = \Omega(2^{\sqrt{2}^{\sqrt{\lg n}}}) = \Theta(\lg(\lg^*(n)))$$

$\leftarrow$ close to 1

5) By the master theorem in rosen,

$$T(n) = 4T(n/3) + n\lg(n)$$

$\hookrightarrow$ $a=4,\ b=3,\ c=1,\ d=1$ and $a > b^c \leftrightarrow 4 > 3^1$

hence $T(n) = \Theta(n^{\log_3(4)})$.

Using annihilators,

let $n = 3^k,\ k = \log_3 n,\ t_k = T(3^k),\ t_{k+1} = T(3^{k+1})$

$$\underbrace{T(3^{k+1}) = 4(T(3^k))}_{(E-4)} + \underbrace{n\log(n)}_{(E-1)}$$

annihilated by: $(E-4)$ $(E-1)$

hence; $T(n) = \alpha 4^k + \beta(1^k) = \alpha 4^k + \beta$

$$T(n) = \alpha 4^{\log_3 n} + \beta$$

$$T(n)\ \cancel{\approx\Theta}\,(3^{(\log_3 4)(\log_3 n)})$$

$$T(n) \approx \Theta(n^{\log_3(4)})$$