Fall 2024 CSCI 605 Assignment #3

Requirement

- **You must print out this assignment template, fill it with your answers in writing, and you must not change the template –** Note: You will lose 5 pts if you fail to do so.
- Scan your assignment into a PDF file and submit it to Canvas.
- **You must scan clearly. Unreadable content will not be graded.**
- Your final PDF file should be named as: "YourFirstName.YourLastName.CSCI-605.A03.pdf". Example: **John.Doe.CSCI-605.A03.pdf –** Note: You will lose 5 pts if your file is not named correctly.
- You can work with other students, but must write up and submit the solutions independently.

Submission

- Due: Sept. 27, 2024, 23:59:59
- PDF or JPG file on Canvas

Problems

0. Full Names

Khushi Choudhary.

1. Solve the following recurrence relations. Be sure to show your work.

$$(a)\ T(n) = \begin{cases} 0 & \text{, if } n \le 1 \\ 2T(\frac{n}{4}) + T(\frac{n}{2}) + n & \text{, if } n > 1 \end{cases}$$

$$(b)\ T(n) = \begin{cases} 1 & \text{, if } n \le 1 \\ 5T(n-5) + 1 & \text{, if } n > 1 \end{cases}$$

(a) Conclusion (claim) (5 pts): T(n) = $O(n \log n)$

Show your work below (10 pts): (You can continue on the next page.)

a) $T(n) = \begin{cases} 0 & \text{if } n \leq 1 \\ 2T\left(\frac{n}{4}\right) + T\left(\frac{n}{2}\right) + n & \text{if } n > 1 \end{cases}$

Substitution Method:-

Guess :- $T(n) = O(n \log n)$

Assume $T(k) \leq ck \log k$ for all $k < n$

for some constant $c > 0$

$$T(n) \leq 2T\left(\frac{n}{4}\right) + T\left(\frac{n}{2}\right) + n$$

Substituting Guess into recurrence →

$$T(n) \leq 2\left(c\frac{n}{4}(\log n - 2)\right) + c\left(\frac{n}{2}(\log(n-1))\right) + n.$$

Simplifying Each Term →

for $T\left(\frac{n}{4}\right)$: $2\left(c\frac{n}{4}(\log n - 2)\right) = \frac{cn}{2}\log n - cn.$

for $T\left(\frac{n}{2}\right)$: $c\left(\frac{n}{2}(\log n - 1)\right) = \frac{cn}{2}\log n - \frac{cn}{2}$

$$T(n) \leq \left(\frac{cn}{2}\log n - cn\right) + \left(\frac{cn}{2}\log n - \frac{cn}{2}\right) + n$$

Combining Terms → $T(n) \leq cn \log n - cn + cn.$

$= cn \log n.$

$T(n) \leq cn \log n$ $T(n) = O(n \log n)$

(b) Conclusion (claim) (5 pts): T(n) = $O(5^n)$

Show your work below (10 pts):

b) $T(n) = \begin{cases} 1 & \text{if } n \leq 1 \\ 5T(n-5)+1 & \text{if } n > 1 \end{cases}$

$T(n) = 5T(n-5)+1$ — ①

$T(n-5) = 5T(n-5-5)+1$

$T(n-5) = 5T(n-10)+1$ — ②

$T(n-10) = 5T(n-15)+1$ — ③

Substitute ② in ①

$T(n) = 5(5T(n-10)+1+1)$

Substitute ③ in ①

$T(n) = 5^2(5T(n-15)+1+2)$

$T(n) = 5^3(T(n-15)+3)$

$T(n) = 5^k(T(n-5k)+c)$

$n-5k = 1$

$n = 1+5k$

$k = \frac{n-1}{5}$

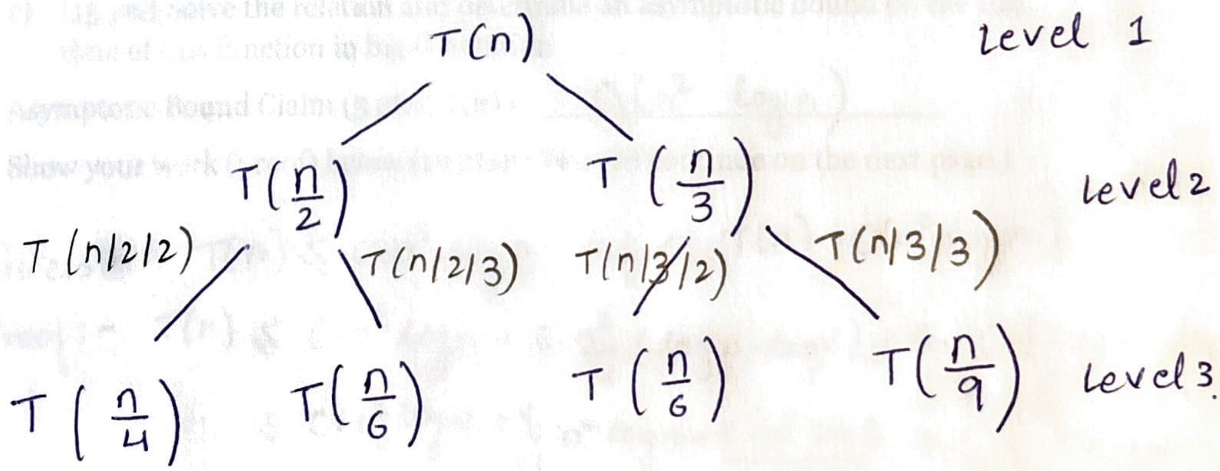
$5^{\frac{n-1}{5}} + \frac{1}{4}(5^{n-1/5}-1)$

Time Complexity → $O(5^n)$

2. Consider the pseudocode below.

```
function f(n)
  if n <= 2
    return 4*n
  k = 1
  for i from 1 to n
    for j from i to n
      k = 2*k+12
  return k + f(n/2) + 3*f(n/3)
```

a) (15 pts) Draw the first three levels (including the root) of a recursive tree for this function. Each node should include T(.). (Level 1 (5 pts), Level 2 (5 pts), Level 3 (5 pts))



b) (25 pts) Write a recurrence relation describing the run time of this function.

Recurrence Relation Claim (10 pts):

$$T(n) = \begin{cases} O(1) & \text{if } n \leq 2 \\ T(\frac{n}{3}) + T(\frac{n}{2}) + n^2 & \text{if } n > 2 \end{cases}$$

Show your work below on how you got this description (hint: sum of atomic steps) (15 pts):

function $f(n)$ if $n \leq 2 \rightarrow O(1)$ $K \times 1 \rightarrow O(1)$

return $4 \times n \rightarrow O(1)$

for i from 1 to $n \rightarrow O(n)$

for j from i to $n \rightarrow O(n)$.

for k from 20 k at 12 $\rightarrow \Theta(1)$

Return $k + f(\frac{n}{2}) + 3 \cdot f(\frac{n}{3}) + f(\frac{n}{2}) f(\frac{n}{3})$

$T(n) = O(1) + O(1) + O(1) + O(1) + O(n) \cdot O(n) + T(\frac{n}{2}) + T(\frac{n}{3})$

$T(n) = T(\frac{n}{3}) + T(\frac{n}{2}) + O(n^2) + 4 \cdot O(1)$

$$T(n) = T(\frac{n}{3}) + T(\frac{n}{2}) + O(n^2)$$

c) (15 pts) Solve the relation and determine an asymptotic bound on the run time of this function in big O notation

Asymptotic Bound Claim (5 pts): T(n) = $O(n^2 \log n)$

Show your work (proof) below (10 pts): (You can continue on the next page.)

Guess: $T(n) \leq c \cdot n^2 \log n$ for $T(n) = O(n^2 \log n)$

Proof:- $T(n) \leq c \cdot n^2 \log n + c \cdot \frac{n^2}{4}(\log n - \log 2) + c \cdot \frac{n^2}{9}(\log n - \log 3)$

$\leq c \cdot n^2 \log n + \frac{c}{4} n^2 \log n - \frac{c \cdot n^2 \log 2}{4} + \frac{c}{9} \cdot n^2 \log n - \frac{c \cdot n^2 \log 3}{9}$

$\leq \frac{c}{4} \cdot n^2 \log n - \frac{c \cdot n^2 \log 2}{4} + \frac{c}{9} \cdot n^2 \log n - \frac{c n^2 \log 3}{9}$

$\Rightarrow n^2 \left(\frac{\log n}{4} + \frac{\log n}{9}\right) - c\frac{\log 2}{4} - \frac{c \cdot \log 3}{9} = 0$

$$\Rightarrow n^2 \left(\frac{13 \log n}{36}\right) = c \left(\frac{\log 2 - 4 \log 3}{36}\right)$$

$$n = \sqrt{\frac{c\,(\log 2 - 4\log 3) \cdot 36}{13 \cdot \log n}}$$

for $c = 1$, $n_0 = \sqrt{\frac{(\log 2 - 4\log 3)}{13}}\, 6$

$$\simeq \sqrt{0.391} * 6 \approx 4$$

Prove $\frac{13c}{36} < 1 \longrightarrow 13c \leq 36$, $c \leq \frac{36}{13}$, $n_0 = 4$

Conclusion $\rightarrow$ $n_0 = 4$ $c = \frac{36}{13}$ $T(n) = O(n^2 \log n)$

3. Given the pseudocode below, show the snapshots of the list L after each round of sorting (bubbling). (Refer to the example on the lecture slides on how to do it, and remember to include the initial list.)

```
function f(L)
  changed = true
  n = length(L)
  while changed
    changed = false
    for i from 1 to n-1
      if L[i-1] > L[i]
        x = L[i-1]
        L[i-1] = L[i]
        L[i] = x
        changed = true
    n = n - 1
  return L
```

a) (5 pts) L = [13, 4, 8, 5, 1] (Correctness: 2 pts; Completeness: 2 pts)

i = 0

13	4	8	5	1
4	13	8	5	1
4	8	13	5	1
4	8	5	13	1
4	8	5	1	13

i = 1

4	8	5	1	13
4	5	8	1	13
4	5	1	8	13

i = 2

| 4 | 5 | 1 | 8 | 13 |
| 4 | 1 | 5 | 8 | 13 |

i = 3

| 4 | 1 | 5 | 8 | 13 |
| 1 | 4 | 5 | 8 | 13 |

L = [13, 4, 8, 5, 1]

Iteration 0 → [4, 8, 5, 1, 13]

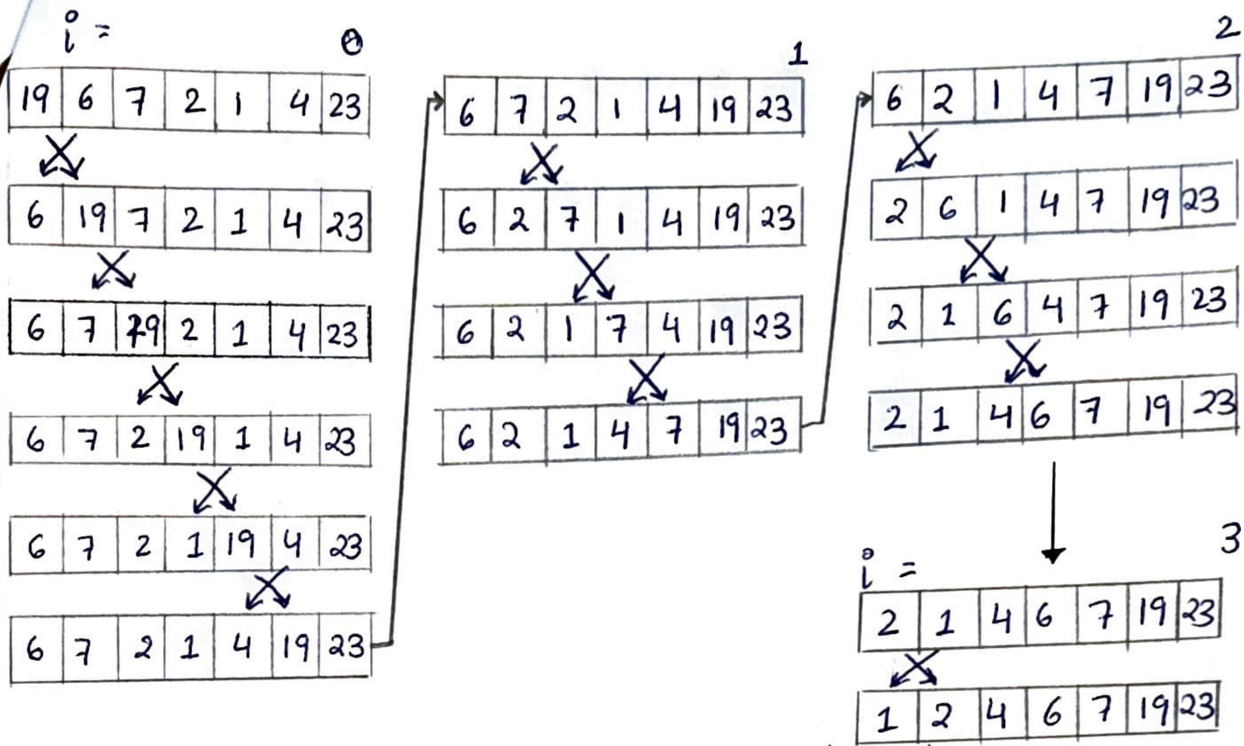
Iteration 1 → [4, 5, 1, 8, 13]

Iteration 2 → [4, 1, 5, 8, 13]

Iteration 3 → [1, 4, 5, 8, 13]

Final Sorted L → [1, 4, 5, 8, 13]

b) (10 pts) L = [19, 6, 7, 2, 1, 4, 23] (Correctness: 6 pts; Completeness: 4 pts)



4. List the people you worked with for this assignment

Zakir Elaskar, Pushpak Rane.

L = [19, 6, 7, 2, 1, 4, 23]

Iteration 0 = [6, 7, 2, 1, 4, 19, 23]

Iteration 1 = [6, 2, 1, 4, 7, 19, 23]

Iteration 2 = [2, 1, 4, 6, 7, 19, 23]

Iteration 3 = [1, 2, 4, 6, 7, 19, 23]

Final Sorted L = [1, 2, 4, 6, 7, 19, 23]