CSCI 3104 Fall 2022
Instructors: Prof. Grochow and Nagesh

Problem Set 6 SOLUTION

Due Date .... October 17, 2022
Name .... **Your Name**
Student ID .... **Your Student ID**
Collaborators .... **List Your Collaborators Here**

Contents

Instructions

- The solutions **must be typed**, using proper mathematical notation. We cannot accept hand-written solutions. Useful links and references on LaTeXcan be found here on Canvas.
- You should submit your work through the **class Canvas page** only. Please submit one PDF file, compiled using this LaTeX template.
- You may not need a full page for your solutions; pagebreaks are there to help Gradescope automatically find where each problem is. Even if you do not attempt every problem, please submit this document with no fewer pages than the blank template (or Gradescope has issues with it).
- You are welcome and encouraged to collaborate with your classmates, as well as consult outside resources. You must **cite your sources in this document. Copying from any source is an Honor Code violation. Furthermore, all submissions must be in your own words and reflect your understanding of the material.** If there is any confusion about this policy, it is your responsibility to clarify before the due date.

- Posting to **any** service including, but not limited to Chegg, Reddit, StackExchange, etc., for help on an assignment is a violation of the Honor Code.
- You **must** virtually sign the Honor Code (see Section Honor Code). Failure to do so will result in your assignment not being graded.

Honor Code (Make Sure to Virtually Sign the Honor Pledge)

Problem HC. On my honor, my submission reflects the following:

- My submission is in my own words and reflects my understanding of the material.
- Any collaborations and external sources have been clearly cited in this document.
- I have not posted to external services including, but not limited to Chegg, Reddit, StackExchange, etc.
- I have neither copied nor provided others solutions they can copy.

In the specified region below, clearly indicate that you have upheld the Honor Code. Then type your name.

Honor Pledge. □

16 Standard 16 - Analyzing Code: Writing Down Recurrences

Problem 16. For each algorithm, write down the recurrence relation for the number of times they print "Welcome". (In this case, this is big-Θ of the runtime - do you see why?) **Don't forget to include the base cases.**

16(a). Problem 16(a)

Algorithm 1 Writing Recurrences 1

(a)

```
procedure Foo(Integer n)
    if n ≤ 5 then return n

    Foo(n/10)
    Foo(n/5)
    Foo(n/5)

    for i ← 1; i ≤ 3 ∗ n; i ← i + 1 do
        print "Welcome"
```

Answer.

$$T(n) = \begin{cases} 0 & : \text{if } n \leq 5, \\ T(n/10) + 2T(n/5) + 3n & : \text{otherwise.} \end{cases}$$

□

16(b). Problem 16(b)

Algorithm 2 Writing Recurrences 2

(b)

```
procedure Foo2(Integer n)
    if n ≤ 7 then return

    Foo2(n/14)
    Foo2(n/14)

    for i ← 1; i ≤ n; i ← i * 2 do
        for j ← 1; j ≤ n; j ← j * 3 do
            print "Welcome"
```

Answer.

$$T(n) = \begin{cases} 0 & : \text{if } n \le 7, \\ 2T(n/14) + \log_3(n)\log_2(n) & : \text{otherwise.} \end{cases}$$

□

17 Standard 17 - Solving Recurrences I: Unrolling

Problem 17. For each of the following recurrences, solve them using the unrolling method (i.e. find a suitable function $f(n)$ such that $T(n) \in \Theta(f(n))$). **Note:** Show all the work.

17(a). Problem 17(a)

a.

$$T(n) = \begin{cases} 3n & : n < 3, \\ 2T(n/2) + 4n & : n \geq 3. \end{cases}$$

Answer.

$$\begin{aligned} T(n) &= 2T(n/2) + 4n \\ &= 2\left(2T(n/4) + 2n\right) + 4n \\ &= 4T(n/4) + 8n \\ &= 4(2T(n/8) + n) + 8n \\ &= 8T(n/8) + 12n \\ &\ldots \\ &= 2^k T(n/2^k) + 4kn. \end{aligned}$$

Solve for when k unrollings will reach the base case:

$$\begin{aligned} n/2^k &< 3 \\ n &< 3 \cdot 2^k \\ \log_2 n &< \log_2(3) + \log_2(2^k) \\ \log_2 n &< \log_2(3) + k \\ k &> \log_2 n - \log_2(3) \\ k &= \lceil \log_2 n - log_2(3) \rceil \end{aligned}$$

Plug back in:

$$\begin{aligned} T(n) &= 2^{\lceil \log_2 n - \log_2 3 \rceil} T(n/2^{\lceil \log_2 n - \log_2 3 \rceil}) + 4(\lceil \log_2 n - \log_2 3 \rceil)n \\ T(n) &\sim n \cdot 3 + 4n \log_2 n + \Theta(1) \\ &= \Theta(n(\log n)) \end{aligned}$$

□

17(b). Problem 17(b)

b.

$$T(n) = \begin{cases} 5 & : n < 2, \\ 7T(n-2) + 9 & : n \geq 2. \end{cases}$$

Answer. Unroll until we see the pattern:

$$\begin{aligned} T(n) &= 7T(n-2) + 9 \\ &= 7(7(T(n-4) + 9) + 9 \\ &= 7^2 T(n-4) + 9(7+1) \\ &= 7^2(7T(n-6) + 9) + 9(7+1) \\ &= 7^3 T(n-6) + 9(7^2 + 7 + 1) \\ &= 7^k T(n-2k) + 9(\sum_{i=0}^{k-1} 7^i) \end{aligned}$$

Solve for when unrolling will reach the base case:

$$\begin{aligned} n - 2k &< 2 \\ \frac{n-2}{2} &< k \end{aligned}$$

Plug back in:

$$\begin{aligned} T(n) &= 7^k T(n-2k) + 9 \left(\sum_{i=0}^{k-1} 7^i \right) \\ &= 7^{\frac{n-2}{2}} T(2) + 9 \sum_{i=0}^{\frac{n-2}{2}-1} 7^i \\ &= 5 \cdot 7^{\frac{n-2}{2}} + 9 \cdot \frac{7^{\frac{n-2}{2}} - 1}{7-1} \\ &= \Theta(7^{n/2}) \end{aligned}$$

□

18 Standard 18 - Divide and Conquer: Counterexamples

Problem 18. Consider the following problem:

> MAX PAIR SUM
> *Input:* A list L of integers
> *Output:* An index $i \in \{1, \ldots, len(L) - 1\}$ such that $L[i] + L[i+1]$ is maximized (that is, such that $L[i] + L[i+1] \geq L[j] + L[j+1]$ for all j), and the value of $L[i] + L[i+1]$

(Note the list here is 1-indexed, so the problem simply does not consider the last element, as it has nothing to pair it with.)

Consider the algorithm below that attempts to solve this problem. **Give an instance** of input (preferably a list of length at most 6) for which it fails to output the correct value for the above problem, and **explain why it fails**.

Algorithm 3 Proposed divide-and-conquer algorithm for the Max Pair Sum problem

1: **procedure** MAXPAIRSUM(List L) $n \leftarrow len(L)$
2:     **if** $n \leq 1$ **then return** ;
3:     **if** $n = 2$ **then return** $(1, L[1] + L[2])$;
4:     $(i1, sum1) \leftarrow \text{MaxPairSum}(L[1..\lfloor n/2 \rfloor])$;
5:     $(i2, sum2) \leftarrow \text{MaxPairSum}(L[\lfloor n/2 \rfloor + 1..n])$;
6:     **if** $i1 \geq i2$ **then**
7:         **return** $(i1, sum1)$;
8:     **else**
9:         **return** $(i2, sum2)$;

Proof. MANY EXAMPLES CAN WORK. HERE IS ONE. $L = [1, 3, 4, 2]$. The consecutive pair sums are $1 + 3 = 4, 3 + 4 = 7, 4 + 2 = 6$, so we see the max occurs at index 2, with value $3 + 4 = 7$. However, the recursive calls work on the first half and separate half separately, that is, $[1, 3]$ and $[4, 2]$, so they will never consider the correct one. The recursive calls each use the base case on line 3, so they return $(1, 4)$ and $(1, 6)$, and thus on line 9 the algorihthm incorrectly returns 6 as the value of the max pair sum.

The issue more generally is that the max pair sum can span across the first and second halves of the list, but the algorithm divides those two to be separate and never recombines to check the value at that middle index.

□