CSCI 3104 Fall 2024
Instructor: Dr. Lijun Chen

Problem Set 2

Due Date .... September 17, 2024
Name .... **Hyma Jujjuru**
Student ID **110743269**
Collaborators ....

Contents

Instructions

- The solutions **should be typed**, using proper mathematical notation. We cannot accept hand-written solutions. Here's a short intro to LaTeX.
- You should submit your work through the **class Gradescope page** only (linked from Canvas). 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). Failure to do so will result in your assignment not being graded.

Honor Code (Make Sure to Virtually Sign) [5 pts]

Problem HC.
- 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.

Agreed (signature here). I agree to the above, Hyma Jujjuru. □

1 Asymptotic Notations

1.1 Problem 1 [15 pts]

Problem 1. Show the following:

- $10(n+1)^2 = \Theta(n^2)$
- $1000\text{logn} = O(\sqrt{n})$
- $10^{20} = \Theta(1)$

Proof.

- $f(n) = 10(n+1)^2, g(n) = \sqrt{n}$
  $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{10(n+1)^2}{n^2} = \lim_{n\to\infty} \frac{10n^2+20n+10}{n^2} = 10$
  Since the limit is a constant, $f(n) = 10(n+1)^2 = \Theta(n^2)$.

- $f(n) = 1000 \log n, g(n) = \sqrt{n}$
  $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{1000 \log n}{\sqrt{n}} = 1000 \cdot \lim_{n\to\infty} \frac{\log n}{\sqrt{n}} = 1000 \cdot \lim_{n\to\infty} \frac{\frac{1}{n\ln(10)}}{\frac{1}{2\sqrt{n}}} = 1000 \cdot \lim_{n\to\infty} \frac{2\sqrt{n}}{n\ln(10)}$
  $= 1000 \cdot \lim_{n\to\infty} \frac{2}{\sqrt{n}\ln(10)} = 1000 \cdot \frac{2}{\ln(10)} \cdot \lim_{n\to\infty} \frac{1}{\sqrt{n}} = 1000 \cdot \frac{2}{\ln(10)} \cdot \frac{1}{\infty} = 1000 \cdot \frac{2}{\ln(10)} \cdot 0 = 0$
  Since the limit is equal to 0, $f(n) = 1000 \log n = O(\sqrt{n})$.

- $f(n) = 10^{20}, g(n) = 1$
  $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{10^{20}}{1} = 10^{20}$
  Since the limit is a constant, $f(n) = 10^{20} = \Theta(1)$.

$\square$

1.2 Problem 2 [15 pts]

Problem 2. Consider the following functions as runtime of different algorithms. **Using Limit Comparison test or other theorems learnt in class**, order the functions from fastest to slowest running time:

Functions: $n!$, 5^n , n^n, $\sqrt{n^{5n+1}}$.

Answer. Order from fastest running time to slowest: $\sqrt{n^{5n+1}} \in O(n^n), n^n \in O(5^n), 5^n \in O(n!)$

Explanation: Using the Limit Comparison Test,
$\lim_{n\to\infty} \frac{n^n}{5^n} = \lim_{n\to\infty} (\frac{n}{5})^n = \lim_{n\to\infty} (e^{n\ln(\frac{n}{5})}) = \lim_{n\to\infty} (e^n \cdot e^{\ln(\frac{n}{5})}) = \lim_{n\to\infty} ((\frac{n}{5})e^n) = \infty$
Since the limit is $\infty, n^n > 5^n$ or n^n has a faster running time than 5^n.

$\lim_{n\to\infty} \frac{\sqrt{n^{5n+1}}}{n^n} = \lim_{n\to\infty} \frac{n^{\frac{5n+1}{2}}}{n^n} = \lim_{n\to\infty} n^{(\frac{5}{2}n+\frac{1}{2}-n)} = \lim_{n\to\infty} n^{\frac{3}{2}n+\frac{1}{2}} = \infty$
Since the limit is ∞, $\sqrt{n^{5n+1}} > n^n$ or $\sqrt{n^{5n+1}}$ has a faster running time than n^n.

Using the ratio test: $\lim_{n\to\infty} \frac{\frac{5^{n+1}}{(n+1)!}}{\frac{5^n}{n!}} = \lim_{n\to\infty} \frac{5^{n+1}}{(n+1)!} \cdot \frac{n!}{5^n} = \lim_{n\to\infty} \frac{5}{n+1} = 0$
Since the limit is 0, 5^n has a faster running time than $n!$. □

1.3 Problem 3 [15 pts]

Problem 3. Consider the following functions as runtime of different algorithms. **Using Limit Comparison test or other theorems learnt in class**, order the functions from fastest to slowest running time:

Functions: $\log_3 n$, $(\log_4 n)^{14/3}$, $\log_4(n^{14/3})$, $n^{1/1000}$

Answer. Order from fastest to slowest running time:
$n^{1/1000} \in O((\log_4 n)^{14/3}), (\log_4 n)^{14/3} \in O(\log_4(n^{14/3})), \log_4(n^{14/3}) \in \Theta(\log_3 n)$

Explanation: Using limit comparison test,

1. $(\log_4 n)^{14/3} > \log_4(n^{14/3})$

$$\lim_{n\to\infty} \frac{(\log_4 n)^{14/3}}{\log_4(n^{14/3})} = \lim_{n\to\infty} (\frac{3}{14} \log_4 n)^{11/3} = \infty$$

Since the limit is ∞, $(\log_4 n)^{14/3}$ has a faster running time than $\log_4(n^{14/3})$ because $\log_4(n^{14/3}) \in O((\log_4 n)^{14/3})$.

2. $\log_4(n^{14/3}) = \log_3 n$

$$\begin{aligned}
\lim_{n\to\infty} \frac{\log_4(n^{14/3})}{\log_3 n} &= \lim_{n\to\infty} \frac{14 \log_4 n}{3 \log_3 n} \\
&= \frac{14}{3} \cdot \lim_{n\to\infty} \frac{\log_4 n}{\log_3 n} \\
&= \frac{14}{3} \cdot \lim_{n\to\infty} \frac{\frac{1}{2\ln(2)x}}{\frac{1}{x\ln(3)}} \\
&= \frac{14}{3} \cdot \lim_{n\to\infty} \frac{\ln(3)}{2\ln(2)} \\
&= \frac{14}{3} \cdot \frac{\ln(3)}{2\ln(2)} \\
&= \frac{7\ln(3)}{3\ln(2)}
\end{aligned}$$

Since the limit is a constant, $\log_4(n^{14/3}) \in \Theta(\log_3 n)$, so $\log_4(n^{14/3})$ has a similar running time to $\log_3 n$.

3. $(\log_4 n)^{14/3} < n^{1/1000}$

$$
\begin{aligned}
\lim_{n\to\infty} \frac{(\log_4 n)^{14/3}}{n^{1/1000}} &= \lim_{n\to\infty} \frac{(\log_4 e)^{14/3}(\ln n^{14/3})}{n^{1/1000}} \\
&= (\log_4 e)^{14/3} \lim_{n\to\infty} \frac{\frac{14}{3}(\ln n^{11/3})n^{-1}}{\frac{1}{1000}n^{-999/1000}} \\
&= (\log_4 e)^{14/3} \frac{14000}{3} \lim_{n\to\infty} \frac{(\ln n)^{11/3}}{n^{1/1000}} \\
&= (\log_4 e)^{14/3} \frac{14000}{3} \lim_{n\to\infty} \frac{\frac{11}{3}(\ln n)^{8/3}n^{-1}}{\frac{1}{1000}n^{-999/1000}} \\
&= (\log_4 e)^{14/3} \frac{14000}{3} \cdot \frac{11000}{3} \lim_{n\to\infty} \frac{(\ln n)^{8/3}n^{-1}}{n^{-999/1000}} \\
&= (\log_4 e)^{14/3} \frac{14000}{3} \cdot \frac{11000}{3} \lim_{n\to\infty} \frac{(\ln n)^{8/3}}{n^{1/1000}} \\
&= (\log_4 e)^{14/3} \frac{14000}{3} \cdot \frac{11000}{3} \lim_{n\to\infty} \frac{\frac{8}{3}(\ln n)^{5/3}n^{-1}}{\frac{1}{1000}n^{-999/1000}} \\
&= (\log_4 e)^{14/3} \frac{14000}{3} \cdot \frac{11000}{3} \cdot \frac{8000}{3} \lim_{n\to\infty} \frac{(\ln n)^{5/3}n^{-1}}{n^{-999/1000}} \\
&= (\log_4 e)^{14/3} \frac{14000}{3} \cdot \frac{11000}{3} \cdot \frac{8000}{3} \lim_{n\to\infty} \frac{(\ln n)^{5/3}}{n^{1/1000}} \\
&= (\log_4 e)^{14/3} \frac{14000}{3} \cdot \frac{11000}{3} \cdot \frac{8000}{3} \lim_{n\to\infty} \frac{\frac{5}{3}(\ln n)^{2/3}n^{-1}}{\frac{1}{1000}n^{-999/1000}} \\
&= (\log_4 e)^{14/3} \frac{14000}{3} \cdot \frac{11000}{3} \cdot \frac{8000}{3} \cdot \frac{5000}{3} \lim_{n\to\infty} \frac{(\ln n)^{2/3}n^{-1}}{n^{-999/1000}} \\
&= (\log_4 e)^{14/3} \frac{14000}{3} \cdot \frac{11000}{3} \cdot \frac{8000}{3} \cdot \frac{5000}{3} \lim_{n\to\infty} \frac{(\ln n)^{2/3}}{n^{1/1000}} \\
&= (\log_4 e)^{14/3} \frac{14000}{3} \cdot \frac{11000}{3} \cdot \frac{8000}{3} \cdot \frac{5000}{3} \lim_{n\to\infty} \frac{\frac{2}{3}(\ln n)^{-1/3}n^{-1}}{\frac{1}{1000}n^{-999/1000}} \\
&= (\log_4 e)^{14/3} \frac{14000}{3} \cdot \frac{11000}{3} \cdot \frac{8000}{3} \cdot \frac{5000}{3} \cdot \frac{2000}{3} \lim_{n\to\infty} \frac{(\ln n)^{-1/3}n^{-1}}{n^{-999/1000}} \\
&= (\log_4 e)^{14/3} \frac{14000}{3} \cdot \frac{11000}{3} \cdot \frac{8000}{3} \cdot \frac{5000}{3} \cdot \frac{2000}{3} \lim_{n\to\infty} \frac{1}{(\ln n)^{1/3}n^{1/1000}} \\
&= (\log_4 e)^{14/3} \frac{14000}{3} \cdot \frac{11000}{3} \cdot \frac{8000}{3} \cdot \frac{5000}{3} \cdot \frac{2000}{3} \cdot 0 \\
&= 0
\end{aligned}
$$

Since the limit is $0, n^{1/1000}$ has a faster running time than $(\log_4 n)^{14/3}$ because $n^{1/1000} \in O((\log_4 n)^{14/3})$.

$\square$

2 Analyze Code I: Independent nested loops

2.1 Problem 4 [15 pts]

Problem 4. Analyze the *worst-case* runtime of the following algorithm:

1. Clearly derive the runtime complexity function $T(n)$ for this algorithm.
2. Using the **formal definition of Big-Theta (Θ)**, find a tight asymptotic bound for $T(n)$, that is, find a function $f(n)$ such that $T(n) = \Theta(f(n))$.

Avoid heuristic arguments from CSCI 2270/2824 such as multiplying the complexities of nested loops.

Algorithm 1 Nested Algorithm 1

```
1: procedure IndependentNested1(Integer n)
2:     for i ← 1;  i ≤ n;  i ← i * 2 do
3:         for j ← 1;  j ≤ n;  j ← j + 2 do
4:             print "Hello"
```

Answer.

1. The inner loop is line 3. It takes one step to initialize and move 1 to i. The loop will end when it runs $1 + 2k > n$ where k stores the value of j each iteration, so when $k > \frac{n-1}{2}$.
   For each iteration, it is checked if $j \leq n$. Then we increment by 2 steps and 2 is added to j, which takes 2 steps, since j is redefining j into $j + 2$. The print statement takes 1 step.
   The j time complexity is

$$\begin{aligned} T(n) &= 1 + \sum_{j=1}^{\frac{n-1}{2}} (1 + 2 + 1) \\ &= 1 + 4(\frac{n-1}{2}) \\ &= 1 + 2n - 2 \\ &= 2n - 1 \end{aligned}$$

   The outer loop is line 2. It takes 1 step to initialize $i = 1$. The loop will end when it runs $i = n + 1$ times. For each iteration, it takes 1 step to check $i \leq n$. Then, we take 2 steps to redefine i as $i * 2$. The loop with run $\log_2 n$ times. Also, we apply the run time for the inner loop (calculated above).
   The combined time complexity is:

$$\begin{aligned} T(n) &= 1 + \sum_{i=1}^{\log_2 n} (1 + 2 + 2n - 1) \\ &= 1 + \sum_{i=1}^{\log_2 n} (2n + 2) \\ &= 1 + (2n + 2)(\log_2 n) \\ &= 2n \log_2 n + 2 \log_2 n + 1 \end{aligned}$$

2. Using the limit comparison test, we show that $T(n) \in \Theta(n \log_2 n)$ with $f(n) = 2n \log_2 n + 2 \log_2 n + 1, g(n) = n \log_2 n$

$$
\begin{aligned}
\lim_{n\to\infty} \frac{2n \log_2 n + 2 \log_2 n + 1}{n \log_2 n} &= \lim_{n\to\infty} (2 + \frac{2}{n} + \frac{1}{n \log_2 n}) \\
&= \lim_{n\to\infty} (2) + \lim_{n\to\infty} (\frac{2}{n}) + \lim_{n\to\infty} (\frac{1}{n \log_2 n}) \\
&= 2 + \frac{2}{\infty} + \frac{1}{\infty} \\
&= 2 + 0 + 0 \\
&= 2
\end{aligned}
$$

Since the limit is a constant, $f(n) = T(n) = 2n \log_2 n + 2 \log_2 n + 1 \in \Theta(n \log_2 n)$

$\square$

3 Analyze Code II: Dependent nested loops

3.1 Problem 5 [15 pts]

Problem 5. Analyze the *worst-case* runtime of the following algorithm:

1. Clearly derive the runtime complexity function $T(n)$ for this algorithm.
2. Using the **formal definition of Big-Theta (Θ)**, find a tight asymptotic bound for $T(n)$, that is, find a function $f(n)$ such that $T(n) = \Theta(f(n))$.

Avoid heuristic arguments from CSCI 2270/2824 such as multiplying the complexities of nested loops.

Algorithm 2 Nested Algorithm 3

```
1: procedure DependentNested1(Integer n)
2:     for i ← 1;  i ≤ n;  i ← i + 1 do
3:         for j ← i;  j ≤ n;  j ← j + 2 do
4:             print "Hola"
```

Answer.

1. The inner loop is line 3. It takes 1 step to initialize $j = i$. It takes 1 step to check $j \leq n$. It takes 2 steps to redefine j as $j + 2$. The loop will end when it runs $1 + 2k > n$ where k stores the value of j each iteration, so when $k > \frac{n-1}{2}$. The j time complexity is:

$$\begin{aligned}
T(n) &= 1 + \sum_{k=1}^{\frac{n-1}{2}} (1 + 2 + 1) \\
&= 1 + \sum_{k=1}^{\frac{n-1}{2}} (4) \\
&= 1 + 4 \cdot (n - i + 1) \\
&= 1 + 4n - 4i + 4 \\
&= 4n - 4i + 5
\end{aligned}$$

The outer loop is line 2. It takes 1 step to initialize $i = 1$. The loop will end when it runs n times.
For each iteration, it takes 1 step to check $i \leq n$. Then, we take 2 steps to redefine i as $i+1$. Also, we apply the run time for the inner loop (calculated above).
The combined time complexity is:

$$\begin{aligned}
T(n) &= 1 + \sum_{i=1}^{n} (1 + 2 + 4n - 4i + 5) \\
&= 1 + \sum_{i=1}^{n} (8 + 4n - 4i) \\
&= 1 + \sum_{i=1}^{n} 8 + \sum_{i=1}^{n} 4n - \sum_{i=1}^{n} 4i \\
&= 1 + 8n + 4n^2 - 4(\frac{n(n+1)}{2}) \\
&= 1 + 8n + 4n^2 - 2n(n+1) \\
&= 1 + 8n + 4n^2 - 2n^2 - 2n \\
&= 2n^2 + 6n + 1
\end{aligned}$$

2. Using the limit comparison test and L'Hôpital, we show that $T(n) \in \Theta(n^2) f(n) = 2n^2 + 6n + 1, g(n) = n^2$

$$\begin{aligned}
\lim_{n\to\infty} \frac{2n^2 + 6n + 1}{n^2} &= \lim_{n\to\infty} \frac{4n + 6}{2n} \\
&= \lim_{n\to\infty} \frac{4}{2} \\
&= 2
\end{aligned}$$

Since the limit is a constant, $f(n) = T(n) \in \Theta(n^2)$.

$\square$

3.2 Problem 6 [25 pts]

Problem 6. Given an array A of n integers, you are asked to calculate the subarray sum $B[i,j]$ (for $i < j$) which is the sum of elements $A[i]$ through $A[j]$ (i.e., the sum $A[i] + A[i+1] + \cdots + A[j]$).

Consider the following simple algorithm to solve this problem:

```
SubArraySum(A) {
   for i=1 to n
      for j = i+1 to n
         B[i,j] = 0
         for k = i to j
             B[i,j] = B[i,j] + A[k]
         end
      end
   end
}
```

1. For some function f that you should choose, give a bound of the form $O(f(n))$ on the running time of this algorithm on an input of size n (i.e., a bound on the number of operations performed by the algorithm).
2. For this same function f, show that the running time of the algorithm on an input of size n is also $\Omega(f(n))$. (This shows an asymptotically tight bound of $\Theta(f(n))$ on the running time.)

Answer. Finding the time complexity of the algorithm:
The innermost loop is: for k = i to j
It takes 1 step to initialize k to i, 1 step to check if $k < j$, and 2 steps to redefine k to $k+1$ because you add 1 to k and then define k to be that value. It takes 2 steps to redefine $B[i,j]$ to $B[i,j] + A[k]$.
The k time complexity is:

$$\begin{aligned}
T(n) &= 1 + \sum_{k=1}^{n}(1+2+2) \\
&= 1 + \sum_{k=1}^{n}(5) \\
&= 1 + 5(n - (j - i) + 1) \\
&= 1 + 5(n - j + i + 1) \\
&= 1 + 5n - 5j + 5i + 5 \\
&= 5n - 5j + 5i + 6
\end{aligned}$$

The middle loop is: for j = i+1 to n
It takes 1 step to initialize $j = i + 1$, 1 step to check if $j \leq n$, and 2 steps to redefine j to $j + 1$. There is also 1 step for the initialization of $B[i,j]$. We also add the time complexity of the innermost loop from above. The loop will end when $j = n$.

Note: Number of iterations: n - 2 + 1 = n-1 The j time complexity is:

$$\begin{aligned}
T(n) &= 1 + \sum_{j=1}^{n-1}(1 + 2 + 1 + 5n - 5j + 5i + 6) \\
&= 1 + \sum_{j=1}^{n-1}(5n - 5j + 5i + 10) \\
&= 1 + \sum_{j=1}^{n-1}(5n) - \sum_{j=1}^{n-1}(5j) + \sum_{j=1}^{n-1}(5i) + \sum_{j=1}^{n-1}(10) \\
&= 1 + 5n(n-1) - 5(n - i - 1 + 1) + (5i)(n-1) + 10(n-1) \\
&= 1 + 5n^2 - 5n - 5n + 5i + 5i(n) - 5i + 10n - 10 \\
&= 5n^2 + 10n + 5i(n) - 9
\end{aligned}$$

The outer loop is: for i=1 to n
It takes 1 step to initialize $i = 1$, 1 step to check if $i \leq n$, 2 steps to redefine i to $i + 1$. We also add the combined time complexity of j and k from above. The loop will end when $i = n + 1$.
The combined time complexity is:

$$\begin{aligned}
T(n) &= 1 + \sum_{i=1}^{n}(1 + 2 + 1 + 5n^2 + 10n + 5i(n) - 9) \\
&= 1 + \sum_{i=1}^{n}(5n^2 + 10n + 5i(n) - 5) \\
&= 1 + \sum_{i=1}^{n}(5n^2) + \sum_{i=1}^{n}(10n) + \sum_{i=1}^{n}(5i(n)) - \sum_{i=1}^{n}(5) \\
&= 1 + 5n^3 + 10n^2 + 5n(\frac{n(n-1)}{2}) - 5n \\
&= 5n^3 + 10n^2 - 5n + 1 + \frac{5}{2}(n^2)(n-1) \\
&= 5n^3 + 10n^2 - 5n + 1 + \frac{5}{2}(n^3 - n^2) \\
&= \frac{5}{2}n^3 + \frac{15}{2}n^2 - 5n + 1
\end{aligned}$$

1. $T(n) = \frac{5}{2}n^3 + \frac{15}{2}n^2 - 5n + 1$. So, $f(n) = n^3$ such that $T(n) \in O(n^3)$.
2. $T(n) = \frac{5}{2}n^3 + \frac{15}{2}n^2 - 5n + 1$. So, $f(n) = n^3$ such that $T(n) \in \Omega(n^3)$.

$\square$