CSCI 3104 Spring 2022
Instructor: Profs. Chen and Layer

Problem Set 4

Due Date .... February 15
Name .... **Julia Troni**
Student ID **109280095**
Collaborators .... **Null**

Contents

1 Instructions

- The solutions **must 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 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 2). Failure to do so will result in your assignment not being graded.

2 Honor Code (Make Sure to Virtually Sign)

Problem 1.

- 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.

I agree to the above, Julia Troni. □

3 Standard 10- Asymptotics I (Calculus I techniques)

Problem 2. For each part, you will be given a list of functions. Your goal is to order the functions from **slowest growing** to **fastest growing**. That is, if your answer is $f_1(n), \dots, f_k(n)$, then it should be the case that $f_i(n) \in O(f_{i+1}(n))$ for all i. If two adjacent functions have the same order of growth (that is, $f_i(n) \in \Theta(f_{i+1}(n))$), clearly specify this. **Show all work, including Calculus details.** Plugging into WolframAlpha is not sufficient.

You may find the following helpful.

- Recall that our asymptotic relations are transitive. So if $f(n) \in O(g(n))$ and $g(n) \in O(h(n))$, then $f(n) \in O(h(n))$. The same applies for Big-Theta, etc. Note that the goal is to order the growth rates, so transitivity is very helpful. We encourage you to make use of transitivity rather than comparing all possible pairs of functions, as using transitivity will make your life easier.
- You may also use the Limit Comparison Test. However, you **MUST** show all limit computations at the same level of detail as in Calculus I-II. Should you choose to use Calculus tools, whether you use them correctly will count towards your score.
- You may **NOT** use heuristic arguments, such as comparing degrees of polynomials or identifying the "high order term" in the function.
- If it is the case that $g(n) = c \cdot f(n)$ for some constant c, you may conclude that $f(n) = \Theta(g(n))$ without using Calculus tools. You must clearly identify the constant c (with any supporting work necessary to identify the constant- such as exponent or logarithm rules) and include a sentence to justify your reasoning.

You may also find it helpful to order the functions using an `itemize` block, with the work following the end of the `itemize` block.

- This function grows the slowest: $f_1(n)$
- These functions grow at the same asymptotic rate and faster than $f_1(n)$: $f_2(n), f_3(n), \dots$
- These functions grow at the same asymptotic rate, but faster than $f_2(n)$: $f_k(n)$.

Also below is an example of an `align` block to help you organize your work.

$$\begin{aligned}
\lim_{n\to\infty} \frac{n^2}{2^n} &= \lim_{n\to\infty} \frac{2n}{\ln(2) \cdot 2^n} \\
&= \lim_{n\to\infty} \frac{2}{(\ln(2))^2 \cdot 2^n} \\
&= 0.
\end{aligned}$$

3.1 Problem 2(a)

(a) $n^3 - 10, \qquad n^3 + 20n^2 + 1000, \qquad n^4 - 50n^3, \qquad 10n^3\sqrt{n}.$

Answer. You may also find it helpful to order the functions using an `itemize` block, with the work following the end of the `itemize` block.

- These functions grow at the same asymptotic rate and they grow the slowest: $f_1(n) = n^3 - 10$, $f_2(n) = n^3 + 20n^2 + 1000$
- This function grows faster than $f_1(n)$ and $f_2(n)$: $f_3(n) = 10n^3\sqrt{n}$
- This function grows the fastest of the 4 equations, faster than $f_1(n)$ and $f_2(n)$ and $f_3(n)$: $n^4 - 50n^3$

I will prove this using the limit comparison test below

(a)

$$\begin{aligned}
\lim_{n\to\infty} \frac{n^3 - 10}{n^3 + 20n^2 + 1000} &= \lim_{n\to\infty} \frac{3n^2}{3n^2 + 40n} \\
&= \lim_{n\to\infty} \frac{6n}{6n + 40} \\
&= \lim_{n\to\infty} \frac{6}{6} = 1
\end{aligned}$$

Thus: $n^3 - 10 = \Theta(n^3 + 20n^2 + 1000)$, so they grow at the same asymptotic rate

(b)

$$\begin{aligned}
\lim_{n\to\infty} \frac{10n^3\sqrt{n}}{n^3 - 10} &= \lim_{n\to\infty} \frac{10n^{\frac{7}{2}}}{n^3 - 10} \\
&= \lim_{n\to\infty} \frac{35n^{\frac{5}{2}}}{3n^2} \\
&= \lim_{n\to\infty} \frac{35 \cdot \frac{5}{2}n^{\frac{3}{2}}}{6n} \\
&= \lim_{n\to\infty} \frac{35 \cdot \frac{5}{2} \cdot \frac{3}{2}n^{\frac{1}{2}}}{6} \\
&= \lim_{n\to\infty} \frac{525n^{\frac{1}{2}}}{24}\infty
\end{aligned}$$

Thus: $10n^3\sqrt{n} = \Omega(n^3 - 10)$ and $(n^3 - 10) = O(10n^3\sqrt{n})$, so $10n^3\sqrt{n}$ grows at a faster rate

(c)

$$\begin{aligned}
\lim_{n\to\infty} \frac{10n^3\sqrt{n}}{n^4-50n^3} &= \lim_{n\to\infty} \frac{10n^{\frac{7}{2}}}{n^4-50n^3} \\
&= \lim_{n\to\infty} \frac{35n^{\frac{5}{2}}}{4n^3-150n^2} \\
&= \lim_{n\to\infty} \frac{35\cdot\frac{5}{2}n^{\frac{3}{2}}}{12n^2-300n} \\
&= \lim_{n\to\infty} \frac{35\cdot\frac{5}{2}\cdot\frac{3}{2}n^{\frac{1}{2}}}{24n-300} \\
&= \lim_{n\to\infty} \frac{35\cdot\frac{5}{2}\cdot\frac{3}{2}\cdot\frac{1}{2}n^{\frac{-1}{2}}}{24} \\
&= \lim_{n\to\infty} \frac{525}{192n^{\frac{1}{2}}} = 0
\end{aligned}$$

Thus: $10n^3\sqrt{n} = O(n^4-50n^3)$ and $(n^4-50n^3) = \Omega(10n^3\sqrt{n})$, so n^4-50n^3 grows at a faster rate

□

3.2 Problem 2(b)

(b) $10 \log_2 n^3, \qquad (\log_3 n)^3, \qquad 100 \log_4 n, \qquad n^{1/1000}.$

Hint: Recall change of logarithmic base formula $\log_a x = \log_b x \cdot \log_a b$

Answer.
- These functions grow at the same asymptotic rate and they grow the slowest: $f_1(n) = 10 \log_2 n^3$, $f_2(n) = 100 \log_4 n$
- This function grows faster than $f_1(n)$ and $f_2(n)$: $f_3(n) = (\log_3 n)^3$
- This function grows the fastest of the 4 equations, faster than $f_1(n)$ and $f_2(n)$ and $f_3(n)$: $f_4(n) = n^{1/1000}$.

I will prove this using logarithm rules and the limit comparison test below

(a)

$$\begin{aligned}
\lim_{n\to\infty} \frac{(\log_3 n)^3}{n^{1/1000}} &= \lim_{n\to\infty} \frac{3 \cdot (\log_3 n)^2 (\frac{1}{n \ln 3})}{\frac{1}{1000} n^{\frac{-999}{1000}}} & &= \lim_{n\to\infty} \frac{3000 (\log_3 n)^2}{\ln 3 \cdot n^{\frac{1}{1000}}} \\
&= \lim_{n\to\infty} \frac{6000 (\log_3 n)(\frac{1}{n \ln 3})}{\ln 3 \cdot \frac{1}{1000} \cdot \frac{1}{1000} n^{\frac{-999}{1000}}} & &= \lim_{n\to\infty} \frac{6,000,000 (\log_3 n)}{(\ln 3)^2 \cdot n^{\frac{1}{1000}}} \\
&= \lim_{n\to\infty} \frac{6,000,000 (\frac{1}{n \ln 3})}{(\ln 3)^2 \cdot \frac{1}{1000} n^{\frac{-999}{1000}}} & &= \lim_{n\to\infty} \frac{6,000,000,000}{(\ln 3)^3 \cdot n^{\frac{1}{1000}}} \\
&= 0
\end{aligned}$$

Thus: $(\log_3 n)^3 = O(n^{1/1000})$ and $n^{1/1000} = \Omega((\log_3 n)^3)$ so $n^{1/1000}$ grows faster than $(\log_3 n)^3$

(b)

$$\begin{aligned}
\lim_{n\to\infty} \frac{10 \log_2 n^3}{(\log_3 n)^3} &= \lim_{n\to\infty} \frac{30 \log_2 n}{(\log_3 n)^3} \\
&= \lim_{n\to\infty} \frac{30 \log_3 n \cdot \log_2 3}{(\log_3 n)^3} \\
&= \lim_{n\to\infty} \frac{30 \log_2 3}{(\log_3 n)^2} \\
&= 0
\end{aligned}$$

Thus: $10 \log_2 n^3 = O(\log_3 n)^3)$ and $(\log_3 n)^3 = \Omega(10 \log_2 n^3)$, so $(\log_3 n)^3$ grows at a faster rate

(c)

$$\begin{aligned}
\lim_{n\to\infty} \frac{10 \log_2 n^3}{100 \log_4 n} &= \lim_{n\to\infty} \frac{30 \log_2 n}{100 \log_4 n} \\
&= \lim_{n\to\infty} \frac{30 \log_2 n}{100 \log_2 n \cdot \log_4 2} \\
&= \lim_{n\to\infty} \frac{3}{10 \log_4 2}
\end{aligned}$$

Thus: $10 \log_2 n^3 = \Theta(100 \log_4 n)$, meaning they grow at the same asymptotic rate.

$\square$

4 Standard 11- Asymptotics II (Calculus II techniques):

Problem 3. *For each of the following questions, put the growth rates in order, from slowest-growing to fastest. That is, if your answer is $f_1(n), f_2(n), \ldots, f_k(n)$, then $f_i(n) \in O(f_{i+1}(n))$ for all i. If two adjacent ones are asymptotically the same (that is, $f_i(n) = \Theta(f_{i+1}(n))$), you must specify this as well. Justify your answer (show your work). You may use transitivity: if $f(n) \in O(g(n))$ and $g(n) \in O(h(n))$, then $f(n) \in O(h(n))$, and similarly for Big-Theta, etc. The same instructions as for Problem 1 apply.*

4.1 Problem 3(a)

(a) 1 , $n^{\log_5 n^2}$, $n^{\log_2 n}$, $n^{\log_n (n^3)}$, $n^{\log_n 10}$.

Answer.

- These functions grow at the same asymptotic rate and they grow the slowest: $f_1(n) = 1$, $f_2(n) = n^{\log_n 10}$
- This function grows faster than $f_1(n)$ and $f_2(n)$: $f_3(n) = n^{\log_n (n^3)}$
- These functions grow the fastest of the 5 functions, faster than $f_1(n)$ and $f_2(n)$ and $f_3(n)$ and they grow at the same asymptotic rate: $f_4(n) = n^{\log_2 n}$ and $f_5(n) = n^{\log_5 n^2}$.

I will prove this using logarithm rules and the limit comparison test below

(a)

$$\begin{aligned}\lim_{n\to\infty} \frac{1}{n^{\log_n 10}} &= \lim_{n\to\infty} \frac{1}{10} \\ &= \frac{1}{10}\end{aligned}$$

Thus: $1 = \Theta(n^{\log_n 10})$, meaning they grow at the same asymptotic rate.

(b)

$$\begin{aligned}\lim_{n\to\infty} \frac{n^{\log_n 10}}{n^{\log_n n^3}} &= \lim_{n\to\infty} \frac{10}{n^3} \\ &= 0\end{aligned}$$

Thus: $n^{\log_n 10} = O(n^{\log_n n^3})$ and $n^{\log_n n^3} = \Omega(n^{\log_n 10})$ so $n^{\log_n n^3}$ grows faster than $n^{\log_n 10}$

(c)

$$\begin{aligned}\lim_{n\to\infty} \frac{n^{\log_n n^3}}{n^{\log_2 n}} &= \lim_{n\to\infty} \frac{n^3}{n^{\log_2 n}} \\ &= \lim_{n\to\infty} \frac{\log_n(n^3)}{\log_n(n^{\log_2 n})} \\ &= \lim_{n\to\infty} \frac{\log_n n^3}{\log_2 n} \\ &= \lim_{n\to\infty} \frac{3}{\log_2 n} \\ &= 0\end{aligned}$$

Thus: $n^{\log_n n^3} = O(n^{\log_2 n})$ and $n^{\log_2 n} = \Omega(n^{\log_n n^3})$ so $n^{\log_2 n}$ grows faster than $n^{\log_n n^3}$

(d)

$$\begin{aligned}
\lim_{n\to\infty} \frac{n^{\log_5 n^2}}{n^{\log_2 n}} &= \lim_{n\to\infty} \frac{\log_n(n^{\log_5 n^2})}{\log_n(n^{\log_2 n})} \\
&= \lim_{n\to\infty} \frac{\log_5 n^2}{\log_2 n} \\
&= \lim_{n\to\infty} \frac{2\log_5 n}{\log_2 n} \\
&= \lim_{n\to\infty} \frac{2\log_2 n \log_5 2}{\log_2 n} \\
&= 2\log_5 2
\end{aligned}$$

Thus: $n^{\log_5 n^2} = \Theta(n^{\log_2 n})$ and $n^{\log_2 n} = \Theta(n^{\log_5 n^2})$, meaning they grow at the same asymptotic rate.

□

4.2 Problem 3(b)

(b) $n!$, 3^n , $3^{n/2}$, n^n, 3^{n-2}, $\sqrt{n^{3n+1}}$. (*Hint:* Recall Stirling's approximation, which says that $n! \sim \left(\frac{n}{e}\right)^n \sqrt{2\pi n}$, i.e. $\lim_{n\to\infty} \frac{n!}{\left(\frac{n}{e}\right)^n \sqrt{2\pi n}} = 1$. We can also say $n! = \Theta(\left(\frac{n}{e}\right)^n \sqrt{2\pi n})$).

Answer.
- This function grows the slowest: $f_1(n) = 3^{n/2}$
- These functions grow faster than $f_1(n)$ and at the same asymptotic rate: $f_2(n) = 3^n$, $f_3(n) = 3^{n-2}$
- This function grows faster than $f_1(n)$ and $f_2(n)$ and $f_3(n)$: $f_4(n) = n!$
- This function grows faster than $f_1(n)$ and $f_2(n)$ and $f_3(n)$ and $f_4(n)$: $f_5(n) = n^n$
- This function grows the fastest of the 6 functions, faster than $f_1(n)$ and $f_2(n)$ and $f_3(n)$ and $f_4(n)$ and $f_5(n)$: $f_6(n) = \sqrt{n^{3n+1}}$

I will prove this using the root test, ratio test, L'Hopital's rule, and the limit comparison test below

(a)

$$\lim_{n\to\infty} \frac{n!}{3^n}$$

$$\begin{aligned}
&\text{Let } a_n = \frac{n!}{3^n} \\
&\text{Using the ratio test we have:} \\
&= \lim_{n\to\infty} \frac{a_{n+1}}{a_n} \\
&= \lim_{n\to\infty} \frac{\frac{(n+1)!}{(3^{n+1})}}{\frac{n!}{3^n}} \\
&= \lim_{n\to\infty} \frac{\frac{(n+1)(n!)}{(3 \cdot 3^n)}}{\frac{n!}{3^n}} \\
&= \lim_{n\to\infty} \frac{n+1}{3} \\
&= \infty
\end{aligned}$$

Thus: $n! = \Omega(3^n)$ and $3^n = O(n!)$, so $n!$ grows faster than 3^n

(b)

$$\lim_{n\to\infty} \frac{3^n}{n^n}$$

$$\begin{aligned}
&\text{Let } a_n = \frac{3^n}{n^n} \\
&\text{Using the root test we have:} \\
&= \lim_{n\to\infty} |a_n|^{\frac{1}{n}} \\
&= \lim_{n\to\infty} |\frac{3^n}{n^n}|^{\frac{1}{n}} \\
&= \lim_{n\to\infty} \frac{3}{n} \\
&= 0
\end{aligned}$$

Thus: $n^n = \Omega(3^n)$ and $3^n = O(n^n)$, so n^n grows faster than 3^n.

(c)

$$\begin{aligned}\lim_{n\to\infty}\frac{n!}{n^n} &= \lim_{n\to\infty}\frac{(\frac{n}{e})^n\sqrt{2\pi n}}{n^n}\\ &= \lim_{n\to\infty}\frac{\sqrt{2\pi}n^{\frac{1}{2}}n^n}{e^n\cdot n^n}\\ &= \lim_{n\to\infty}\frac{\sqrt{2\pi}n^{\frac{1}{2}}}{e^n}\\ &= \lim_{n\to\infty}\frac{\sqrt{2\pi}\cdot\frac{1}{2}\cdot n^{\frac{-3}{2}}}{e^n}\\ &= \lim_{n\to\infty}\frac{\sqrt{2\pi}}{2e^n\cdot n^{32}}\\ &= 0\end{aligned}$$

Thus: $n^n = \Omega(n!)$ and $n! = O(n^n)$, so n^n grows faster than $n!$

(d)

$$\lim_{n\to\infty}\frac{3^n}{3^{\frac{n}{2}}}$$

$$\begin{aligned}&= \lim_{n\to\infty}3^{n-\frac{n}{2}}\\ &= \lim_{n\to\infty}3^{\frac{n}{2}}\\ &= \infty\end{aligned}$$

Thus: $3^n = \Omega(3^{\frac{n}{2}})$ and $3^{\frac{n}{2}} = O(3^n)$, so 3^n grows faster than $3^{\frac{n}{2}}$

(e)

$$\begin{aligned}\lim_{n\to\infty}\frac{3^n}{3^{n-2}} &= \lim_{n\to\infty}3^{n-(n-2)}\\ &= \lim_{n\to\infty}3^2\\ &= 9\end{aligned}$$

Thus: $3^n = \Theta(3^{n-2})$ and $3^{n-2} = \Theta(3^n)$, meaning they grow at the same asymptotic rate.

(f)

$$\lim_{n\to\infty}\frac{n^n}{\sqrt{n^{3n+1}}}$$

Let $a_n = \dfrac{n^n}{n^{\frac{3n+1}{2}}}$

Using the root test we have:

$$\begin{aligned}&= \lim_{n\to\infty}|a_n|^{\frac{1}{n}}\\ &= \lim_{n\to\infty}\left|\frac{n^n}{n^{\frac{3n+1}{2}}}\right|^{\frac{1}{n}}\\ &= \lim_{n\to\infty}\frac{n}{n^{\frac{3}{2}}\cdot n^{\frac{1}{2n}}}\\ &= \lim_{n\to\infty}\frac{n}{n^{\frac{3n+1}{2n}}}\\ &= \lim_{n\to\infty}n^{1-\frac{3n+1}{2n}}\\ &= \lim_{n\to\infty}n^{\frac{-1}{2}-\frac{1}{2n}}\\ &= \lim_{n\to\infty}\frac{1}{n^{2+2n}}\\ &= 0\end{aligned}$$

Thus: $\sqrt{n^{3n+1}} = \Omega(n^n)$ and $n^n = O(\sqrt{n^{3n+1}})$, so $\sqrt{n^{3n+1}}$ grows faster than n^n

$\square$

5 Standard 12- Analyzing Code I: (Independent nested loops)

Problem 4. Analyze the *worst-case* runtime of the following algorithms. Clearly derive the runtime complexity function $T(n)$ for this algorithm, and then find a tight asymptotic bound for $T(n)$ (that is, find a function $f(n)$ such that $T(n) \in \Theta(f(n))$). Avoid heuristic arguments from 2270/2824 such as multiplying the complexities of nested loops.

```
1: procedure foo_i(integer n):
2:   for i = 1, i <= n
3:       i = 3 * i
4:       print ‘outer’
5:       for j = 1, j <= n
6:           j = 2 * j
7:           print ‘inner’
```

Answer. I begin by analyzing the inner j loop:

- At line 5, the j loop takes 1 step to initialize j=1
- The loop terminates when $2^k > n$. Solving for k, we have that: $k > log_2 n$. The loop takes at least one iteration to compare j to n, so the loop takes $\lfloor log_2 n \rfloor + 1$ iterations
- At each iteration, the loop does the following:
  - The comparison $j <= n$ takes 1 step
  - The update $j = 2 * j$ takes 2 steps: 1 step to evaluate $2 * j$ and 1 step for the assignment
  - The body of the loop consists of a single print statement, which takes 1 step.

  So the run time complexity of the j loop is :

$$\begin{aligned} 1 + \sum_{j=1}^{\lfloor log_2 n \rfloor + 1} (1 + 2 + 1) &= 1 + \sum_{j=1}^{\lfloor log_2 n \rfloor + 1} (4) \\ &= 1 + 4(\lfloor log_2 n \rfloor + 1) \\ &= 5 + 4\lfloor log_2 n \rfloor \end{aligned}$$

Now analyze the outer loop:

- Initializing the loop at line 2 takes 1 step
- The loop terminates when $3^k > n$. Solving for k, we have that: $k > log_3 n$. The loop takes at least one iteration to compare i to n, so the loop takes $\lfloor log_3 n \rfloor + 1$ iterations
- At each iteration, the loop does the following:
  - The comparison $i <= n$ takes 1 step
  - The update $i = 3 * i$ takes 2 steps: 1 step to evaluate $3 * i$ and 1 step for the assignment
  - The body of the loop consists of print statement, which takes 1 step and the inner j loop.

  So the run time complexity function is:

$$\begin{aligned} T(n) = 1 + \sum_{i=1}^{\lfloor log_3 n \rfloor + 1} ((1 + 2 + 1) + (5 + 4\lfloor log_2 n \rfloor)) &= 1 + \sum_{i=1}^{\lfloor log_3 n \rfloor + 1} (9) + \sum_{i=1}^{\lfloor log_3 n \rfloor + 1} (4\lfloor log_2 n \rfloor) \\ &= 1 + 9(\lfloor log_3 n \rfloor + 1) + 4((\lfloor log_3 n \rfloor + 1) \cdot \lfloor log_2 n \rfloor) \\ &= 1 + 9 + 9\lfloor log_3 n \rfloor + 4((\lfloor log_3 n \cdot log_2 n \rfloor) + 4\lfloor log_2 n \rfloor) \\ &= 10 + 9\lfloor log_3 n \rfloor + 4(\lfloor log_3 n \cdot log_2 n \rfloor) + 4\lfloor log_2 n \rfloor \end{aligned}$$

So the run time complexity is $T(n) = \Theta(\log_2 n \cdot log_3 n)$

$\square$

6 Standard 13- Analyzing Code II: (Dependent nested loops)

Problem 5. Analyze the *worst-case* runtime of the following algorithms. Clearly derive the runtime complexity function $T(n)$ for this algorithm, and then find a tight asymptotic bound for $T(n)$ (that is, find a function $f(n)$ such that $T(n) \in \Theta(f(n))$). Avoid heuristic arguments from 2270/2824 such as multiplying the complexities of nested loops.

```
1: procedure foo_d(integer n):
2:   for i = 1, i <= n
3:       i = i + 1
4:       print ‘outer’
5:       for j = 1, j <= i
6:           j = j + 2
7:           print ‘inner’
```

Answer. I begin by analyzing the inner j loop:

- At line 5, the j loop takes 1 step to initialize j=1
- The j loop takes $\lfloor \frac{i}{2} \rfloor + 1$ iterations. At each iteration, the loop does the following:
  - The comparison $j <= i$ takes 1 step
  - The update $j = j + 2$ takes 2 steps: 1 step to evaluate $j + 2$ and 1 step for the assignment
  - The body of the loop consists of a single print statement, which takes 1 step.

  So the run time complexity of the j loop is :

$$\begin{aligned} 1 + \sum_{j=1}^{\lfloor \frac{i}{2} \rfloor + 1} (1 + 2 + 1) &= 1 + \sum_{j=1}^{\lfloor \frac{i}{2} \rfloor + 1} (4) \\ &= 1 + 4(\lfloor \frac{i}{2} \rfloor + 1) \\ &= 5 + 2i \end{aligned}$$

Now analyze the outer loop:

- Initializing the loop at line 2 takes 1 step
- The i loop takes n iterations. At each iteration, the loop does the following:
  - The comparison $i <= n$ takes 1 step
  - The update $i = i + 1$ takes 2 steps: 1 step to evaluate $i + 1$ and 1 step for the assignment
  - The body of the loop consists of print statement, which takes 1 step and the inner j loop.

  So the run time complexity function is:

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

So the run time complexity is $T(n) = \Theta(n^2)$

□