

Name:
ID:

CSCI 3104, Algorithms
Homework 1B (70 points)

Escobedo & Jahagirdar
Summer 2020, CU-Boulder

Advice 1: For every problem in this class, you must justify your answer: show how you arrived at it and why it is correct. If there are assumptions you need to make along the way, state those clearly.

Advice 2: Verbal reasoning is typically insufficient for full credit. Instead, write a logical argument, in the style of a mathematical proof.

Instructions for submitting your solution:

- The solutions **should be typed**, we cannot accept hand-written solutions. Here's a short intro to [Latex](#).
- In this homework we denote the asymptomatic *Big-O* notation by \mathcal{O} and *Small-O* notation is represented as o .
- We recommend using online Latex editor [Overleaf](#). Download the **.tex** file from Canvas and upload it on overleaf to edit.
- You should submit your work through [Gradescope](#) only.
- If you don't have an account on it, sign up for one using your CU email. You should have gotten an email to sign up. If your name based CU email doesn't work, try the identikey@colorado.edu version.
- Gradescope will only accept **.pdf** files (except for code files that should be submitted separately on Canvas if a problem set has them) and **try to fit your work in the box provided**.
- You cannot submit a pdf which has less pages than what we provided you as Gradescope won't allow it.

Piazza threads for hints and further discussion

Piazza Threads

[Question 1](#)

[Question 2](#)

Recommended reading: For complete background read Chapters 1, 2 and 3. Chapter 3 will especially be helpful.

1. ($4 \times 10 = 40$ pts) For each part of this question, put the growth rates in order, from slowest-growing to fastest. That is, if your answer is $f_1(n), f_2(n), \dots, f_k(n)$, then $f_i(n) \leq \mathcal{O}(f_{i+1}(n))$ 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.

(a) Polynomials.

$n^{\frac{1}{2}}, n + 10, n^{\frac{1}{3}}, n^3, n^3 + n^2 + 100, 2^{100}$

(b) Logarithms and related functions.

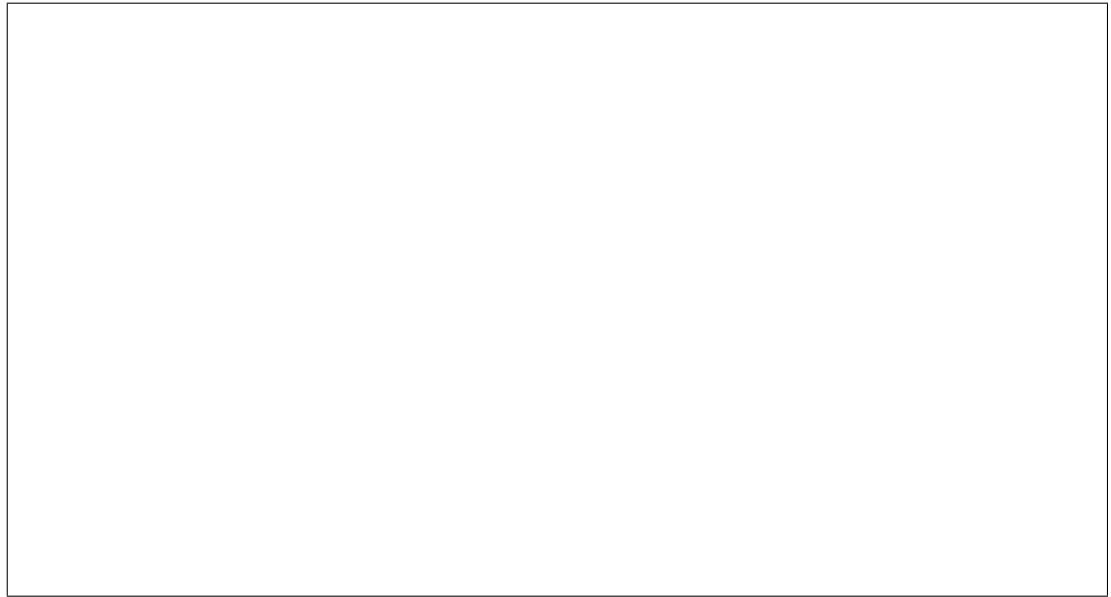
$\log_3 n^2, (\log_3 n)^3, \log_3 n, \log_5 n^2, \log_2 n, \sqrt{n}$

Name:

ID:

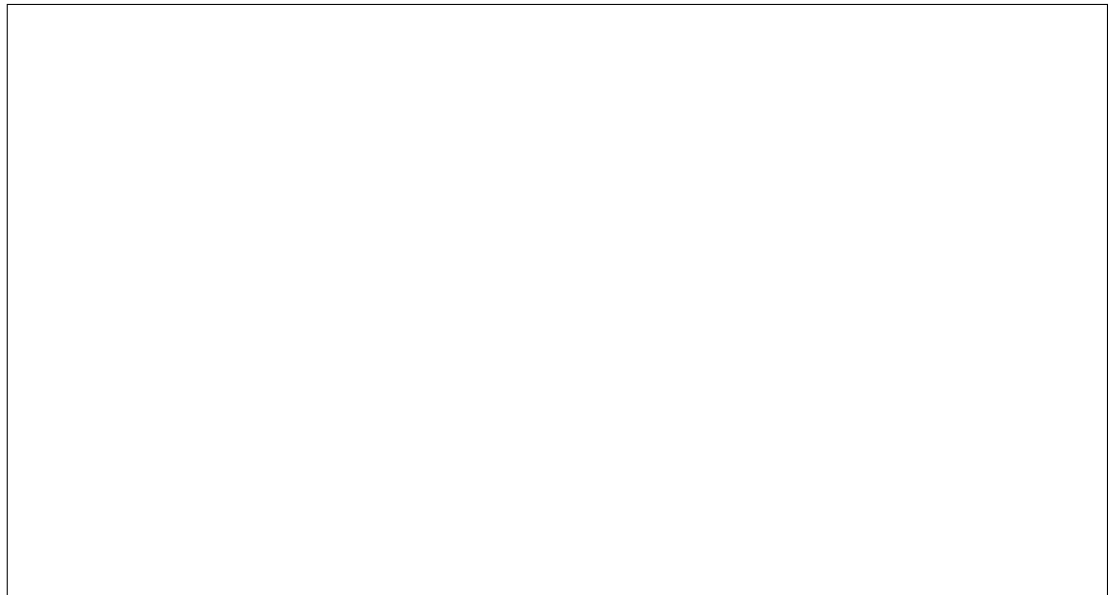
CSCI 3104, Algorithms
Homework 1B (70 points)

Escobedo & Jahagirdar
Summer 2020, CU-Boulder



(c) Logarithms in exponents.

$$n^{\log_3 n}, n^{\frac{1}{\log_3 n}}, n^{\log_4 n}, 1, n$$



(d) Exponentials. (hint: Recall [Stirling's approximation](#), which says that $n! \sim (\frac{n}{e})^n \sqrt{2\pi n}$
i.e. $\lim_{n \rightarrow \infty} \frac{n!}{(\frac{n}{e})^n \sqrt{2\pi n}} = 1$)

Name:

ID:

CSCI 3104, Algorithms
Homework 1B (70 points)

Escobedo & Jahagirdar
Summer 2020, CU-Boulder

$n!, n^5, 2^{2n}, 2^{2n+7}, 5^{n \log_5(n)}$

2. ($3 \times 10 = 30$ pts) For each of the following algorithms, analyze the worst-case running time. You should give your answer in \mathcal{O} notation. You do not need to give an input which achieves your worst-case bound, but you should try to give as tight a bound as possible.

Justify your answer (show your work). This likely means discussing the number of atomic operations in each line, and how many times it runs.

(a)

```
1 count = 0
2 for(i = 1; i < n; i = i + 1)
3 {
4     for(j = i; j < n; j = j + 1)
5     {
6         count = count+1
7     }
8 }
```

Name:

ID:

CSCI 3104, Algorithms
Homework 1B (70 points)

Escobedo & Jahagirdar
Summer 2020, CU-Boulder

(b)

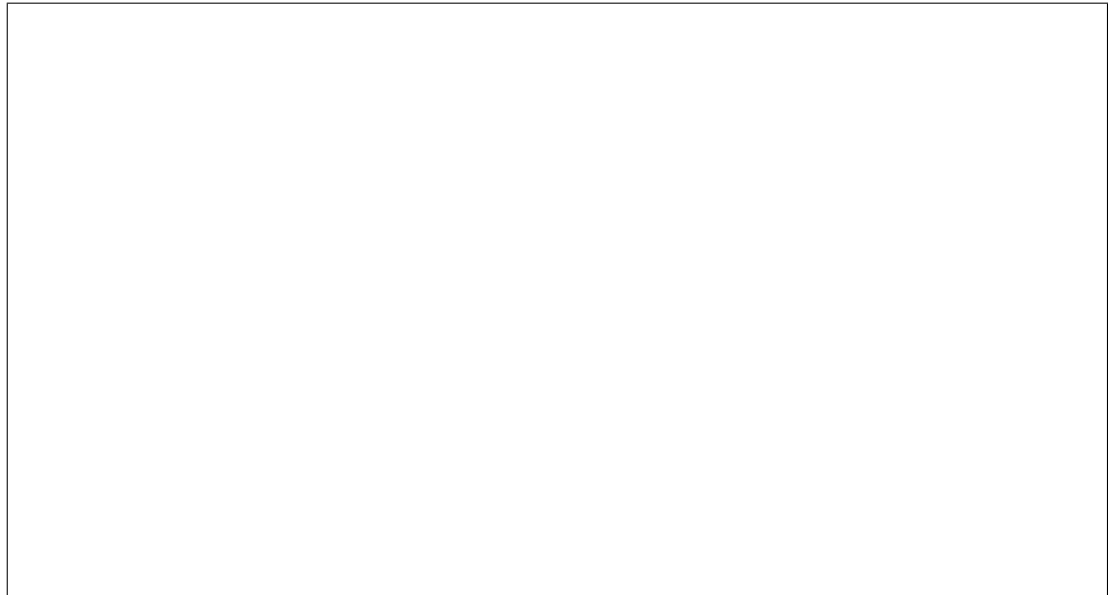
```
1 count = 0
2 for(i = 1; i <= n; i = i * 2)
3 {
4     for(j = 0; j < n; j = j + 2)
5     {
6         count = count+1
7     }
8 }
```

Name:

ID:

CSCI 3104, Algorithms
Homework 1B (70 points)

Escobedo & Jahagirdar
Summer 2020, CU-Boulder



- (c) Here A is a list of integers of size at least 2 and $\text{sqrt}(n)$ returns the square root value of its argument. You can assume that the upper bound of calculating square root takes big $\mathcal{O}(k)$ time. Provide an upper bound in terms of n and k .

```
1  count = 0
2  for(i = 0; i < n; i = i + 1)
3  {
4      for(j = i+1; j < n; j = j + 1)
5      {
6          if(A[i]>sqrt(A[j]))
7          {
8              count = count+1
9          }
10     }
11 }
12 }
```

CSCI 3104, Algorithms
Homework 1B (70 points)

Name:
ID:
Escobedo & Jahagirdar
Summer 2020, CU-Boulder

3. **Extra Credit (5% of total homework grade)** For this extra credit question, please refer the leetcode link provided below or click [here](#). Multiple solutions exist to this question ranging from brute force to the most optimal one. Points will be provided based on Time and Space Complexities relative to that of the most optimal solution. Please provide your solution with proper comments which carries points as well.

<https://leetcode.com/problems/product-of-array-except-self/>

Replace this text with your source code inside of the .tex document