

UNIVERSITY OF HOUSTON

HOMEWORK 0 SOLUTIONS

COSC 3320

Algorithms and Data Structures

Gopal Pandurangan

Read the [University of Houston Academic Honesty policy](#).

Academic Honesty Policy

All submitted work should be your own. Copying or using other people's work (including from the Web) will result in $-\text{MAX}$ points, where MAX is the maximum possible number of points for that assignment. Repeat offenses will result in a failing grade for the course and will be reported to the Chair. If you have any questions, please reach out to the professor and the TAs. The best way to ask is on [Piazza](#).

By submitting this assignment, you affirm that you have followed the Academic Honesty Policy.

Your submission **must be typed**. We prefer you use \LaTeX to type your solutions — \LaTeX is the standard way to type works in mathematical sciences, like computer science, and is highly recommended; for more information on using \LaTeX , please see [this post on Piazza](#) — but any method of typing your solutions (e.g., MS Word, Google Docs, Markdown) is acceptable. **Your submission must be in pdf format**. The assignment can be submitted **up to two days late for a penalty of 10% per day**. A submission more than **two days late** will receive a **zero**.

Reading

Chapters 2, 3, and Appendix A. In particular, several worked exercises with solutions are provided at the end of each chapter. Attempting to solve the worked exercises **before** seeing their solutions is a good learning technique.

The exercises below are from [the book](#). The book is updated periodically, so be sure to use the latest version.

Exercises

2.3, 2.9, 2.12, A.3 (appendix A)

Justify your answers. Show appropriate work.

This page intentionally left blank.

1 Class Questions

Each class question carries 2 points each.

1.1 January 19

Question 1

Show that, in the Celebrity Problem, there can be at most **one** celebrity.

Solution. Suppose there exist two or more celebrities and say A and B are different celebrities. Since A is a celebrity, B must know A . Then B is not a celebrity. This is a contradiction. \square

Question 2

Give a simple lowerbound for the number of questions asked for the Celebrity Problem.

Solution. Clearly, we must ask each person at least one question for a trivial lowerbound of n questions. \square

Question 3

Improve the algorithm given in class for the Celebrity Problem so that it requires $3n - 4$ questions.

Solution. When the remaining candidate, X , is left, we must determine if the other $n - 1$ people know X . However, the last person to be removed has *already been confirmed* to know X , so this person need not be asked. Thus, we can reduce the number of questions asked by 1 to $3n - 4$ in total. \square

Question 4

Prove that, in the $2^n \times 2^n$ Tiling Problem, no solution exists using L-shaped 3-tiles if there is not a “hole” in the board.

Solution. Notice that, if k tiles can fill the grid, then we must have $2^n \times 2^n = 3k$. Then 3 divides 2^{2n} , which is clearly impossible. \square

1.2 January 21

Question 1

Using mathematical induction, show the correctness of the Decrease-and-Conquer algorithm for the Celebrity Problem given in class.

Solution. The base case is trivial. Suppose we have that our algorithm works on $n - 1$ people. Then on n people, we eliminate a single person by the method described in the algorithm, so that $n - 1$ remain. By our induction hypothesis, this algorithm correctly identifies the celebrity candidate amongst those $n - 1$ people. Then the final step of checking whether the $n - 1$ eliminated candidates know our celebrity candidate correctly establishes whether they are a celebrity. \square

Question 2

Explain why an $\mathcal{O}(\sqrt{n})$ algorithm for determining if an integer n is prime is inefficient.

Solution. Algorithms with numerical inputs have *the number of digits (bits) used to represent them as their input size*¹. Then $b = \lceil \log n \rceil$ is our input size, and this algorithm has runtime $\mathcal{O}(\sqrt{n}) = \mathcal{O}(\sqrt{2^b}) = \mathcal{O}(2^{\frac{b}{2}})$, which is *exponential* in the input size. \square

2 Textbook Exercises

Exercise 2.3

(10 points) The algorithm `BinaryISqrt` is iterative. Modify it to make it a recursive algorithm.

Algorithm `BinaryISqrt` – Binary Search Square Root

```

1: def BINARYISQRT( $n$ ):
    Input  ▷ A natural number  $n$ 
    Output ▷  $\lfloor \sqrt{n} \rfloor$ 
2:   low  $\leftarrow 1$                                 ▷ Low value of guess
3:   high  $\leftarrow n$                                ▷ High value of guess
4:   while True:
5:       mid  $\leftarrow \lfloor (\text{low} + \text{high})/2 \rfloor$       ▷ current estimate
6:       if  $\text{mid}^2 \leq n$  and  $(\text{mid} + 1)^2 > n$ :
7:           return mid
8:       else if  $\text{mid}^2 < n$ :                             ▷  $\text{mid} < \lfloor \sqrt{n} \rfloor$ 
9:           low  $\leftarrow \text{mid}$ 
10:      else:                                             ▷  $\text{mid} > \lfloor \sqrt{n} \rfloor$ 
11:          high  $\leftarrow \text{mid}$ 

```

Solution. (10 points for the correct pseudocode.)

Algorithm `RecBinaryISqrt` – Recursive Binary Search Square Root

```

1: def RECBINARYISQRT( $n$ ):
    Input  ▷ A natural number  $n$ 
    Output ▷  $\lfloor \sqrt{n} \rfloor$ 
2:   def HELPER(low, high):                             ▷ Helper function
3:       mid  $\leftarrow \lfloor (\text{low} + \text{high})/2 \rfloor$       ▷ current estimate
4:       if  $\text{mid}^2 \leq n$  and  $(\text{mid} + 1)^2 > n$ :
5:           return mid
6:       else if  $\text{mid}^2 < n$ :                             ▷  $\text{mid} < \lfloor \sqrt{n} \rfloor$ 
7:           return HELPER(mid, high)
8:       else:                                             ▷  $\text{mid} > \lfloor \sqrt{n} \rfloor$ 
9:           return HELPER(low, mid)
10:  return HELPER( $n, 1, n$ )

```

\square

Exercise 2.9

(25 points) There are n adults in town A and they all need to go to town B. There is only a motorbike available which is owned by two boys. The motorbike can carry only one adult or up to two boys at a time (note that at one least person is needed to ride a bike). Using the motorbike, all the adults need to reach town B from town A. Show how this can be accomplished while, at the end, leaving the motorbike with the two boys in town A.

Show the correctness of your algorithm by using mathematical induction and analyze the number of trips needed. (Hint: Use decrease and conquer to reduce the problem size.)

¹Strictly speaking, this can be any base greater than 1, though we typically use base-2 and note that all logarithm bases greater than 1 are asymptotically equal up to a constant factor.

Solution. To begin, have both boys drive the motorbike to town B . Then, have *one* boy drive the bike back to town A . From here, an adult in town A can drive the bike to town B , and the boy who remained in town B can drive the bike back to A . Repeat n times. (10 points for the algorithm)

We will show by induction that the above algorithm transports *all* adults to town B and leaves the two boys in town A . The base case is clear. Suppose that this holds for $n - 1$ adults. Then, for n adults, the above algorithm transports one adult and leaves both boys in town A . Then, with $n - 1$ adults left, the process is repeated $n - 1$ times, the correctness of which follows from the induction hypothesis. (7 points for correctness proof)

Similarly, we note that the number of trips, $T(n)$, is given by $T(n) = T(n - 1) + 4$ with $T(1) = 4$. The base case is given, and again a simple induction argument shows that $T(n) = 4n$:

$$\begin{aligned} T(n) &= T(n - 1) + 4 \\ &= 4(n - 1) + 4 \text{ by the induction hypothesis} \\ &= 4n - 4 + 4 \\ &= 4n \end{aligned}$$

□

(8 points for analysis of trips)

Exercise 2.12

(25 points) Rank the following functions by order of growth; that is, find an arrangement g_1, g_2, g_3, \dots of the functions satisfying $g_1 = \mathcal{O}(g_2)$, $g_2 = \mathcal{O}(g_3), \dots$

$$n^{1.5} + n, \frac{3n}{\log^3 n}, n \log^2 n, 1.01^n, \log^5 n, \frac{1}{n^2}, 4^{\lg n}, n!, (\lg n)^{\lg n}$$

Note: \lg or \log means logarithm to the base 2 and $\log^5 n$ is the usual way of writing $(\log n)^5$. Justify your ordering with arguments.

Solution. (9 points for the correct ordering; 16 points for the justification, 2 each for justifying a consecutive pair correctly.)

The correct ordering is

$$\frac{1}{n^2}, \log^5 n, \frac{3n}{\log^3 n}, n \log^2 n, n^{1.5} + n, 4^{\log n}, \log n^{\log n}, 1.01^n, n!$$

We will show this one-by-one.

- $\frac{1}{n^2} = \mathcal{O}(\log^5 n)$

This follows directly from the limit definition

$$\lim_{n \rightarrow \infty} \frac{\frac{1}{n^2}}{\log^5 n} = \lim_{n \rightarrow \infty} \frac{1}{n^2 \log^5 n} = 0$$

- $\log^5 n = \mathcal{O}\left(\frac{3n}{\log^3 n}\right)$

This follows directly from the limit definition, applying L'Hôpital's Rule

$$\begin{aligned}
\lim_{n \rightarrow \infty} \frac{\log^5 n}{\frac{3n}{\log^3 n}} &= \lim_{n \rightarrow \infty} \frac{\log^8 n}{3n} \\
&= \frac{1}{\ln^8 2} \lim_{n \rightarrow \infty} \frac{\ln^8 n}{3n} \\
&= \frac{1}{\ln^8 2} \lim_{n \rightarrow \infty} \frac{8 \ln^7 n}{3n} \\
&= \frac{1}{\ln^8 2} \lim_{n \rightarrow \infty} \frac{8 \cdot 7 \ln^6 n}{3n} \\
&= \frac{1}{\ln^8 2} \lim_{n \rightarrow \infty} \frac{8 \cdot 7 \cdot 6 \ln^5 n}{3n} \\
&\vdots \\
&= \frac{1}{\ln^8 2} \lim_{n \rightarrow \infty} \frac{8!}{3n} \\
&= 0
\end{aligned}$$

- $\frac{3n}{\log^3 n} = \mathcal{O}(n \log^2 n)$

This follows directly from the limit definition

$$\lim_{n \rightarrow \infty} \frac{\frac{3n}{\log^3 n}}{n \log^2 n} = \lim_{n \rightarrow \infty} \frac{3}{\log^5 n} = 0$$

- $n \log^2 n = \mathcal{O}(n^{1.5} + n)$

This follows directly from the limit definition, applying L'Hôpital's Rule

$$\begin{aligned}
\lim_{n \rightarrow \infty} \frac{n \log^2 n}{n^{1.5} + n} &= \frac{1}{\ln^2 2} \lim_{n \rightarrow \infty} \frac{n \ln^2 n}{n^{1.5} + n} \\
&= \frac{1}{\ln^2 2} \lim_{n \rightarrow \infty} \frac{\ln^2 n + 2 \ln n}{1.5n^{0.5} + 1} \\
&= \frac{1}{\ln^2 2} \lim_{n \rightarrow \infty} \frac{\frac{2 \ln n + 2}{n}}{0.75n^{-0.5}} \\
&= \frac{1}{\ln^2 2} \lim_{n \rightarrow \infty} \frac{\frac{2 \ln n + 2}{n}}{0.75n^{-0.5}} \\
&= \frac{8}{3 \ln^2 2} \lim_{n \rightarrow \infty} \frac{\ln n + 1}{n^{0.5}} \\
&= \frac{8}{3 \ln^2 2} \lim_{n \rightarrow \infty} \frac{2}{n^{0.5}} \\
&= 0
\end{aligned}$$

- $n^{1.5} + n = \mathcal{O}(4^{\log n})$

Note that $4^{\log n} = n^2$. Then, this follows directly from the limit definition:

$$\begin{aligned}
\lim_{n \rightarrow \infty} \frac{n^{1.5} + n}{n^2} &= \lim_{n \rightarrow \infty} \frac{1.5n^{0.5} + 1}{2n} \\
&= \lim_{n \rightarrow \infty} \frac{0.75n^{-0.5} + 1}{2} \\
&= 0
\end{aligned}$$

- $4^{\log n} = \mathcal{O}(\log n^{\log n})$

Note that $4^{\log n} = n^2$ and $\log n^{\log n} = n^{\log \log n}$. Clearly, then, when $\log \log n \geq 2$, $\log n^{\log n} \geq 4^{\log n}$, which holds for $n \geq 2^{2^2} = 2^4 = 16$. Alternatively, this follows from the limit definition.

- $\log n^{\log n} = \mathcal{O}(1.01^n)$

We will compare the logarithms of these functions:

$$\begin{aligned} \lim_{n \rightarrow \infty} \frac{\log n \log \log n}{n \log 1.01} &= \frac{1}{\log 1.01 \ln^2 2} \lim_{n \rightarrow \infty} \frac{\ln n \ln \ln n - \ln n \ln \ln 2}{n} \\ &= \frac{1}{\log 1.01 \ln^2 2} \lim_{n \rightarrow \infty} \frac{\ln \ln n + 1 - \ln \ln 2}{n} \\ &= \frac{1}{\log 1.01 \ln^2 2} \lim_{n \rightarrow \infty} \frac{1}{n \ln n} \\ &= 0 \end{aligned}$$

- $1.01^n = \mathcal{O}(n!)$

This can be proven by induction. The base case is $n = 2$, which is straightforward: $1.01^2 = 1.0201 < 2 = 2!$. Suppose that $1.01^{n-1} < (n-1)!$ for some $n > 3$, then

$$\begin{aligned} 1.01^n &= 1.01 \cdot 1.01^{n-1} \\ &< 1.01 \cdot (n-1)! \text{ by the induction hypothesis} \\ &< n \cdot (n-1)! \text{ since } n > 3 \\ &= n! \end{aligned}$$

More generally, we can show that $a^n < n!$ for sufficiently large n . Consider the series $\sum_{i=0}^{\infty} \frac{a^n}{n!}$ and apply the ratio test:

$$\begin{aligned} \lim_{n \rightarrow \infty} \frac{\frac{a^{n+1}}{(n+1)!}}{\frac{a^n}{n!}} &= \lim_{n \rightarrow \infty} \frac{a}{n} \\ &= 0 \end{aligned}$$

Thus, the series converges. In particular, this forces $\lim_{n \rightarrow \infty} \frac{a^n}{n!} = 0$.

□

Exercise A.3

(10 points) Prove the following statement by mathematical induction:

$$\sum_{i=0}^n a^i = \frac{a^{n+1} - 1}{a - 1}$$

where $a \neq 1$ is some fixed real number.

By the way, this is the sum of a geometric series, a useful formula that one comes across often in algorithmic analysis.

Solution. (base case: 2 points: induction hypothesis and proof: 8 points).

The base case, $n = 0$, is trivial:

$$\sum_{i=0}^0 a^i = a^0 = 1$$

and

$$\frac{a^{0+1} - 1}{a - 1} = \frac{a - 1}{a - 1} = 1$$

Suppose this holds for $n = k - 1$. Then, for $n = k$, we have

$$\begin{aligned}
\sum_{i=0}^k a^i &= \sum_{i=0}^{k-1} a^i + a^k \\
&= \frac{a^{k-1+1} - 1}{a - 1} + a^k \text{ by the induction hypothesis} \\
&= \frac{a^k - 1}{a - 1} + a^k \left(\frac{a - 1}{a - 1} \right) \\
&= \frac{a^k - 1 + a^k(a - 1)}{a - 1} \\
&= \frac{a^k - 1 + a^{k+1} - a^k}{a - 1} \\
&= \frac{a^{k+1} - 1}{a - 1}
\end{aligned}$$

which completes the proof. □