CS 252: Algorithms
Problem Set #1

Layla Oesper, Jeff Ondich
Spring 2024

Due Dates: Your full answers to all questions are due by **5pm on Friday March 29** and will be graded for correctness and clarity.

At the top of your write-up for each problem, please estimate the amount of time you spent on the problem, list your collaborators, and briefly describe the nature of your collaboration. Also, provide a list of other sources you used, including links for online resources. Thanks!

This assignment is intended to be a refresher of content you saw in CS 111, CS 201 and CS 202/Math 236 that will be useful in this class. I encourage you to look back at notes and other resources from those classes as you complete this assignment.

1. (5 points) In previous courses, you likely used big-O notation to describe the time complexity of different operations using various implementations. Informally, we say that an operation (or algorithm) has a time complexity of $O(n)$ if its time requirement is proportional to n. Similarly, we say that an operation (or algorithm) has a time complexity of $O(n^2)$ if its time requirement is proportional to n^2. If the time complexity of an operation (or algorithm) doesn't depend on the problem size n, then we say it has a time complexity of $O(1)$.

   Identify the big-O time complexity for the following operations and implementations. Give a 1-2 sentence explanation along with each answer.

   (a) Retrieve element at position i from an Array of length n.

   (b) Compute the sum of an Array of integers of length n.

   (c) Add an element to a Stack stored using a linked list with a head pointer.

   (d) Remove the least recently inserted element from a Queue stored using a linked list with head and tail pointers.

   (e) Remove the lowest priority item from a Priority Queue stored using a Min-Heap.

2. (2 points) Sorting can often be very useful when designing new algorithms. Since we are interested in designing efficient algorithms, we will often use Merge Sort when we need to sort a list of items. Give a brief description of Merge Sort and identify its big-O running time in terms of the number of items n in the list to be sorted.

3. (4 points) We will focus on designing efficient algorithms this term. It will therefore be important for you to identify how many operations a piece of pseudo-code performs.

   (a) In terms of the variable n, count how many times line 4 of the following piece of pseudo-code gets executed (e.g., your answer might be in a form like $2n+3$)? Give a 1-2 sentence explanation with your answer.

```
1 counter = 0
2 for i = 1,...,n:
3     for j = 1,...,i:
4         counter = counter + 2
```

(b) In terms of the variable n, count how many times the variable `var` gets assigned a value. Give a 1-2 sentence explanation with your answer.

```
var = 1
for i = 1,...,n:
    var = var * i

counter = n
while counter > 0:
    var = var + counter
    counter = counter - 2
```

4. (4 points) At various times this term we will be writing proofs. Often these proofs will allow us to show that our algorithms really do solve the problems we are interested in. You may want to spend some time reviewing proof techniques you have seen in previous classes. One type of proof that we will see repeatedly throughout the term is proof by induction. Below is a Claim we are interested in proving by induction and the start of a proof of that Claim. Your job is to finish writing the proof.

Claim 1. $\sum_{i=1}^{n}(2i-1) = n^2$, *where* $n \in \mathbb{Z}^{\geq 1}$.

Proof. We will prove Claim 1 using induction on the variable n. To do that, we must first show a base case and then an inductive case.

Base case. $(n = 1)$ When $n = 1$ we can show the following series of equalities.

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

Thus we have shown that when $n = 1$, we have $\sum_{i=1}^{n}(2i-1) = n^2$.

Inductive case. $(n \geq 2)$. We start by assuming the inductive hypothesis. That is, we assume the following: $\sum_{i=1}^{n-1}(2i-1) = (n-1)^2$. We now want to use this, in order to show that $\sum_{i=1}^{n}(2i-1) = n^2$.

Your job is to finish writing this proof by induction by completing the Inductive case.

Submission Logistics

Your solutions to each question in this problem set, and all assignments for this course, must be written up and typeset in LaTeX. You must then generate a PDF of your solution which you will upload to Moodle.

This will be the default way to submit homework this term. Occasionally, I might ask for a different kind of submission (e.g., a Python program), but most of the time, you'll do your write-ups in LaTeX.

Please use the class LaTeX template as the basis for your write-up.

Please see the class LaTeX resources page for more LaTeX help.

Need Help?

Don't hesitate to use Slack #questions to ask a question. Chances are someone else is having the same problem and will also benefit from your asking.