Competitive Programming
Lesson 2 – Time Complexity + PSA

Written by Jason Sun and Puneet Bhat

Homework Take-up

Junior

1) Hello, World! – https://dmoj.ca/problem/helloworld

2) A Plus B – https://dmoj.ca/problem/aplusb

3) CCC '14 J2 – Vote Count – https://dmoj.ca/problem/ccc14j2

Senior

1) CCC '14 J1 – Triangle Times – https://dmoj.ca/problem/ccc14j1

2) CCC '23 J3 – Special Event – https://dmoj.ca/problem/ccc23j3

3) UACC 1 P2 – Puzzling Parks – https://dmoj.ca/problem/uacc1p2

1

Time Complexity

How can we measure speeds?

Time?

What is time complexity?

It is a way for us to generalize and easily understand the **speed** of an algorithm or task.

We want to do things fast, so we don't **exceed the time limit**, which is put to disallow slower, unintended/incorrect implementations from passing. To show time complexity, we use something called Big O notation.

Common Examples

O(1)
Constant Time

Example
- Doing simple arithmetic operations
- Accessing a specific element in an array through indices

O(n)
Linear Time

Example
- Finding the min or max of an array
- Finding the sum of an entire array
- Checking string equality

O(n²)
Quadratic Time

Example
- Finding all unique pairs in an array
- Finding the fastest
- Comparing elements in two different arrays

O(log n)
Logarithmic Time

The fastest speed for:
- Binary search (to appear in the future)
- Storing a number as a string

(REFER BACK)

What is the time complexity of the optimal solution to this problem from our last lesson?

Your friend Bob needs help counting.
He gives you an integer **n**.
Output the integers from 1 to **n**, all on different lines, in ascending order.

Output needs to be exact. Example on the right.

INPUT
4

OUTPUT
1
2
3
4

(KNOWLEDGE)

What is the time complexity of the code below?

```
1    n = int(input())
2
3    for a in range(n):
4        for b in range(n):
5            for c in range(n):
6                print(a, b, c)
7
8
```

(THINKING)

What is the time complexity of the code below?

```python
def count_even_sum_pairs(arr):
    count = 0
    n = len(arr)
    for i in range(n):
        for j in range(i + 1, n):
            if (arr[i] + arr[j]) % 2 == 0:
                count += 1
    return count
```

Operations

You can kinda use time complexity to more accurately tell if your code will pass or not.

Most judges do 1 billion operations per second

So if your code runs in $O(n^2)$ time and the constraints are
$0 <= n <= 1000$, meaning n will only be going up to 1000

In the worse case: total time = n^2 = 1000^2 = 1,000,000
Since you will be doing 1 million operations in total, and that is less than one billion, it will probably be enough!

(Given that the time limit is 1 second, but if say the time limit was 2 seconds, then you need to be under 2 billion operations).

2

Prefix Sum Arrays

Putting your new time complexity knowledge to the test! Showing you the benefit of considering time complexity?

Question

Given a list of numbers, and some queries of a ranged sum, print the sum of that range.

Example

Given [4, 1, 0, –2, 5, 9, 3]

And asked (inclusive range) (0–indexed)
-   Sum of index 3 to 3?
-   Sum of index 1 to 4?
-   Sum of index 0 to 6?

But we can actually solve this faster! The easiest solution is too slow.

If I gave you an array of **1 million numbers**, then every query we answer will take **O(n)** time since you have to go through each number and sum them up! This is okay for one query, but if they ask for many queries then that will take too long.

Total Operations = Going Through Every Query * Time it Takes to Find Sum per Query

So what's the time complexity of our solution?

O(n²)

But for the same problem, we can actually improve it to O(n) time.

<u>P</u>refix <u>S</u>um <u>A</u>rray

Given array **A** = [2, 3, 4, 1]
The PSA of **A** would be [2, 5, 9, 10]

In the PSA
  - Index 0 is the sum of **A**'s element 0
  - Index 1 is the sum of **A**'s elements 0, 1
  - Index 2 is the sum of **A**'s elements 0, 1, 2
  - Index 3 is the sum of **A**'s elements 0, 1, 2, 3 (the sum of the whole array)

Awesome! The PSA will help us, and you can start thinking about how using the PSA will let us instantly find the sum of a range in a list. But first off, how do you make it?

Well you could just sum up 0 to 0 for the first index, 0 to 1 for the second index, until your done. But that takes $O(n^2)$, which is too long, the same long time as our previous solution.

Creating the PSA

We can actually use the PSA to create itself!

What do I mean, well, if we want to calculate **psa**[i]
- **psa**[i] = **psa**[i−1] + **A**[i]

A = [2, 3, 4, 1]
psa = [2] (we start with the first element of **A**)

To find **psa**[1] we can add [2, 3, 4, 1] = 2 + 3 = 5
psa = [2, 5]

To find **psa**[2] we can add [2, 3, 4, 1] = 5 + 4 = 9
psa = [2, 5, 9]

And so on.

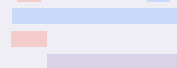Prefix Sum Array and Range Sum Queries
By Profound Academy
https://www.youtube.com/watch?v=xbYr9JOC2Lk

Using the PSA

Sum of range calculation is: blue – red = result (sum)

   A =   [2, 3, 4, 1]

psa = [0, 2, 5, 9, 10] (tip: add 0 at the start so you don't need to check for out of bounds cases)
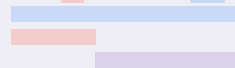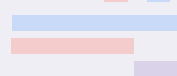
Sum of index 0 to 2:
[0, 2, 5, 9, 10]

Sum of index 1 to 3:
[0, 2, 5, 9, 10]

Sum of index 2 to 2:
[0, 2, 5, 9, 10]

More Information on PSA! Check it out if you want :D

https://usaco.guide/silver/prefix-sums

Practice

If you want to improve, practice! Do the homework! If you want to be really good, do some DMOJ questions outside of class/homework too. You can also take competitive programming classes to keep you accountable.

Homework

Junior (~5p)

1)   Warm Up – Total Sum – https://dmoj.ca/problem/nccc3j1

2)   CCC '24 J3 – Bronze Count – https://dmoj.ca/problem/ccc24j3

3)   Lil' Jami – https://dmoj.ca/problem/liljami

4)   DMOPC '14 Contest 2 P4 – https://dmoj.ca/problem/dmopc14c2p4

Senior (~7p)

1)   CCC '24 J4 – Troublesome Keys – https://dmoj.ca/problem/ccc24j4

2)   James's Egirl Discord Status – https://dmoj.ca/problem/bsspc21s3

3)   Crayola Lightsaber – https://dmoj.ca/problem/bfs17p2

4)   OTHS Mock CCC J4 – Railgun – https://dmoj.ca/problem/othscc1p4

Thanks!

Do you have any questions?

Jason Sun – 350702866@gapps.yrdsb.ca
Puneet Bhat – 349068395@gapps.yrdsb.ca

https://github.com/CryoJS