

CSCI-163 Theory of Algorithms HW1

Tushar Shrivastav

January 23, 2023

1 Problem 1

Problem: Suppose we have an algorithm with the number of basic operations $C(n) = 2^n$, where n is the input size. In the year 2000, we run this program with an input size of $n = 10$, and in the year 2020, our input size grows to 50. Imagine that computer processing speeds roughly double every two years. Which one of the following is correct? Explain your answer by showing your work.

A. The program will run much faster in 2020 with $n = 50$ compared to the year 2000 with $n = 10$ because of the increase in processing speed.

B. The program will take much longer in 2020 with $n = 50$ even though computers are faster.

C. The running time of the program will be approximately the same in both years.

Solution: The correct answer is B. The program will take much longer in 2020 with $n = 50$ even though computers are faster. Since the computer processing speeds increase roughly double every two years, this means a computer in 2020 will be 1024 times faster than a computer in 2000. Using the following equation we can calculate the running time of the two computers for the same algorithm and respective input sizes.

$$T(n) = t_{op} \cdot C(n) \quad (1)$$

Let us assume for the year 2000:

$$t_{op} = 1ms$$

Then for

$$2000 : T(n) = 1 \cdot 2^{10} = 1024$$

$$2020 : T(n) = 1/1024 \cdot 2^{50} \approx 1.0995 \cdot 10^{12}$$

Thus, the running time for 2020 is much greater, regardless of the increase in processing speed.

2 Problem 2

Problem: List the following functions according to their order of growth from the lowest to the highest. Show your work using limits for comparing orders of growth.

$$n^6 + 5n + 1, \log_2(n^3), \sqrt[3]{n} + 10$$

Solution: We can calculate 3 limits to tell us the order of growth from lowest to highest.

$$\lim_{n \rightarrow \infty} \frac{\log_2(n^3)}{n^6 + 5n^5 + 1} = \lim_{n \rightarrow \infty} \frac{3}{\ln(2) \cdot n \cdot (6n^5 + 25n^4)} = \frac{1}{\infty} = 0$$

$$\lim_{n \rightarrow \infty} \frac{\sqrt[3]{n} + 10}{n^6 + 5n^5 + 1} = \lim_{n \rightarrow \infty} \frac{1}{3n^{\frac{2}{3}}(6n^5 + 25n^4)} = \frac{1}{\infty} = 0$$

$$\lim_{n \rightarrow \infty} \frac{\log_2(n^3)}{\sqrt[3]{n} + 10} = \lim_{n \rightarrow \infty} \frac{9n^{\frac{2}{3}}}{\ln(2) \cdot n} = \lim_{n \rightarrow \infty} \frac{9}{\ln(2)n^{\frac{1}{3}}} = 0$$

Thus, Lowest to Highest order of growth is:

$$\log_2(n^3) \rightarrow \sqrt[3]{n} + 10 \rightarrow n^6 + 5n + 1$$

3 Problem 3

Problem: Find a closed-form formula for $\sum_{i=k}^n (3i + n)$. Your answer should include both n and k .

Solution: Expand the summation using summation rules

$$\begin{aligned}
 & 3 \sum_{i=k}^n i + n \sum_{i=k}^n 1 \\
 & 3 \left[\sum_{i=k}^n i - \sum_{i=0}^{k-1} i \right] + n(n - k + 1) \\
 & 3 \left[\frac{n(n+1)}{2} - \frac{k(k-1)}{2} \right] + n(n - k + 1) \\
 & \boxed{\frac{3}{2} [n^2 + n - k^2 + k] + n(n - k + 1)}
 \end{aligned}$$

4 Problem 4

Problem: Consider the following algorithm.

Algorithm 1: Mystery(A[0..n-1, 0..n-1])

Input: A matrix A[0..n-1, 0..n-1] of real numbers

```

1 x = 0
2 for i = 0 to n - 2 do
3   for j = i + 1 to n - 1 do
4     if A[i, j] < A[j, i] then
5       x = x + 1
6 return x

```

- What is the basic operation?
- Count the worst-case number of basic operations executed.
- What is the efficiency class of this algorithm?

Solution:

- The basic operation of this algorithm is $A[i, j] < A[j, i]$
- The worst-case, best-case, and average-case for this algorithm are all the same. The for loops will run until the size of the matrix (N).

Calculate with summation rules:

$$\begin{aligned}
 & \sum_{i=0}^{n-2} \sum_{j=i+1}^{n-1} 1 \\
 & \sum_{i=0}^{n-2} (n - 1 - i - 1 + 1) \\
 & \sum_{i=0}^{n-2} (n - 1 - i) \\
 & n \sum_{i=0}^{n-2} 1 - \sum_{i=0}^{n-2} i \\
 & n \sum_{i=0}^{n-2} 1 - \sum_{i=0}^{n-2} i - \sum_{i=0}^{n-2} i
 \end{aligned}$$

$$n(n-2+1) - (n-2+1) - \left(\frac{(n-2)(n-1)}{2}\right)$$

$$n^2 - 2n + 1 - \left(\frac{n^2 - 3n + 2}{2}\right)$$

$$\frac{2n^2 - 4n + 2 - n^2 + 3n - 2}{2}$$

$$\boxed{\frac{n^2 - n}{2}}$$

c) The algorithm runs in the quadratic efficiency class (n^2).