Computing Complexity of Algorithms

Example Problems

1. Algorithm A runs in $O(N^2)$ time, and for an input size of 4, the algorithm runs in 10 milliseconds, how long can you expect it to take to run on an input size of 16?

$$\frac{4^2}{10} = \frac{16^2}{x}$$

$$\Rightarrow \quad x = 100\text{ms}$$

2. Algorithm A runs in $O(\log_2 N)$ time, and for an input size of 16, the algorithm runs in 28 milliseconds,

how long can you expect it to take to run on an input size of 64?

$$\frac{\log 16}{28 \text{ ms}} = \frac{\log 64}{x} \Rightarrow x = 42 \text{ ms}$$

3. Algorithm A runs in $O(N^3)$ time. For an input size of 10, the algorithm runs in 7 milliseconds.

For another input size, the algorithm takes 189 milliseconds. What was that input size?

$$\frac{10^3}{7 \text{ ms}} = \frac{N^3}{189} \Rightarrow N = 30$$

4. An algorithm takes 6 ms for input size N = 10,
24.9 ms for N = 20,
and 149 ms for N = 50.

Find whether the complexity of the algorithm is $O(N^2)$ or $O(N^3)$.

5. What is the computational complexity of the following algorithm? (Assume N is a positive integer)

```
int N,j,k,sum = 0;

scanf("%d",&N);

j = N;

while (j > 1) {

    k = 0;

    while (k > N){

       sum = sum + j * k;
      k ++;

    }
   j = j / 2;
}
```

The number of operations can be found by a recurrence relation

$T(N) = T(N/2) + 3$

$= T(N/4) + 3 + 3$

$= T(N/8) + 3 (3)$, which can be rewritten as

$= T(N/2^3) + 3 (3)$

$= \ldots\ldots\ldots$

$= T(N/2^k) + k (3)$

Let $2^k = N$, then $k = \log N$

$= T(1) + 3 \log N$

So the complexity turns out to be $O(\log N)$

6. What is the computational complexity of the following algorithm?

```
int N,j,k,sum = 0;

scanf(“%d”,&N);

for(j=1; j <= N; j++){

   for(k=1; k <= j; k++) {

      sum = sum + j * k;

   }
}
```

Two operations are carried out each time the “k” loop is executed. The k loop runs for “j” times.

“j” loop runs N times

For N = 1 , j loop runs 1 time and k loop runs 1 time

For N = 2 , j loop runs 2 times and k loop runs 1+2 times

For N = 3 , j loop runs 3 times and k loop runs 1+2+3

times

For $N = 4$, j loop runs 4 times and k loop runs 1+2+3+4 times

For any value of N, the number of operations are

$2(1 + 2 + 3 + 4 + \ldots\ldots\ldots.. + N)$

$= 2 N(N+1) / 2$

$= N^2 + N$

Thus the order of complexity is $O(N^2)$

Summations

To find the total number of operations in a program involving simple variables, arrays or linked lists, and being computed within a set of loops,

find the number of times the loop is getting executed and the number of operations within the loop.

The number of operations can be found with the help of summations.

There can be a single loop or nested loops. Start with the innermost loop and work outwards.

Let us consider various cases below:

Consider a single loop with a single operation. The summation with the loop variable moving from 1 to n would be given by

$$\Sigma\, 1 = 1 + 1 + 1 + \ldots + n$$
$$= n$$

$$\Sigma\, 4 = 4\ (\Sigma\, 1\ ) = 4n$$

When the number of operations are not fixed, and depend on the loop variable, then the summation can take the following form

$\Sigma\, j \quad = 1 + 2 + 3 + 4 + \ldots\ldots.. + n$

$\qquad = n(n+1)/2$

$\Sigma\, 2j \quad = 2\, \Sigma\, j$

$\qquad = n\,(n+1)$

Now consider two nested loops

$$\sum_{k=1}^{10} \sum_{j=1}^{6} j = \sum^{10} 6(7)/2$$

$$= \sum^{10} 21$$

$$= 21 \sum^{10} 1$$

$$= 210$$

If the summation was from 0 to 10 instead from 1 to 10,
then we would have the result as 21 (11) = 232

If the summation was from –2 to 10 then the result would be 21(13).

If there are 3 operations, with one of them dependent on the loop variable, the summation can take the following form

$$\sum_{i=1} (i+2)$$

Consider the summation as i varies from 1 to 20, where i is also the loop variable.

Decompose the summation into two separate summations:

$$\sum i \quad + \sum 2$$

$$= 20(21)/2 \quad + 2\ (20)$$

If you want to compute summation $\sum (j + 2)$, as j varies from 10 to 20,with j being the loop variable, then you can solve the problem by decomposing the problem in two parts.

First find the summation from 1 to 20, and then subtract summation 1 to 9 from it.

$$\sum_{j=1}^{j=20} (j + 2) = \sum_{j=1}^{j=20} (j + 2) - \sum_{j=1}^{j=9} (j + 2)$$

Finally, summation of a square term in k, with k varying from 1 to n is given by

$$\sum k^2 = n(n+1)(2n + 1) / 6$$

logarithms

$a^b = m$

Taking log of both sides with respect to base a,

$b \log_a a = \log_a m$

or

$b = \log_a m$

Thus knowing any two quantities, you can find the 3rd one.

Base conversion:

Convert 75.75 decimal into binary and octal form.

Convert 200 decimal into binary form.

Practice Problems from an old question paper

1) (3 pts.) Solve for x.
$x^3 = 155 - \log_4 256 * \log_3 81$

Answer: ________________

Hint:

2) (3 pts.)What is the decimal representation of the binary number 10110001?

Answer: ________________

3) (3 pts.) What is the binary representation of the decimal number 472?

Answer:_________________

4) (8 pts.)Calculate the following summations:

a) $\sum_{i=0}^{99} (2i - 15)$

Answer:________________

b) $\sum_{i=10}^{80} (i + 5)$

Answer:__________________

c) $\sum_{i=1}^{100} \sum_{j=10}^{20} 4i$

Answer:__________________