Chapter 3
Algorithms

Section 3.2 :Growth of Functions

Introduction

- The time required to solve a problem depends on-
  - The number of operations it uses.
  - The hardware and software used to run the program that implements the algorithm.
- However, with the change of the hardware and software used to implement an algorithm for a problem of size n can be approximated by-
  - $Time_{new} = Time_{previous} \times c$, where c is a constant.
  - For example, on a supercomputer a problem of size n can be solved $a\ million\ times$ faster than on a PC.
  - However, this factor of one million will not depend on n (except perhaps in some minor ways).

big-O Notation

- big-O Notation:
  - Used to estimate the growth of a function.
    - i.e. To estimate the number of operations an algorithm uses as its input grows.
  - Advantages-
    - Don't have to worry about –
      - The Hardware and the Software used in the implementation.
      - Constant multipliers or smaller order terms.
    - Can assume different operations in an algorithm take the same time and simplify analysis considerably.
  - Application-
    - Determine whether an algorithm is suitable for solving a particular problem as its input grows or not.
    - Compare 2 algorithms and determine the most efficient one with increasing inputs.

big-O Notation (Contd.)

- Definition 1:
  - Let f and g be functions from the set of $integers$ or the set of $real\,numbers$ to the set of $real\,numbers$. We say that $f(x)$ is $O\big(g(x)\big)$[read as "$f(x)$ is $big-oh$ of $g(x)$"] if there are constants C and k, such that $|f(x)| \leq C|g(x)|$, whenever $x > k$.
  - Intuitively, the definition that $f(x)$ is $O\big(g(x)\big)$ says that, $f(x)$ grows slower than some fixed multiple of $g(x)$ as x grows without bound.
  - The constants C and k are called **witnesses** to the relation $f(x)$ is $O\big(g(x)\big)$.
    - We need to find only 1 $pair$ of witnesses C and k, such that $|f(x)| \leq C|g(x)|$, whenever $x > k$.
    - If there is 1 $pair$ of witnesses, then there are infinitely many pairs as well.
    - If C and k are a pair of witnesses, then any pair C' and k', where $C < C'$ and $k < k'$, is also a pair of witnesses as $|f(x)| \leq C|g(x)| \leq C'|g(x)|$ whenever $x > k' > k$.

big-O Notation (Contd.)

- Finding witnesses-
  - Firstly, select a value of k for which the size of $|f(x)|$ can be readily estimated when $x > k$.
  - Secondly, use this estimate to find a value of C for which $|f(x)| \leq C|g(x)|$ for $x > k$.

big-O Notation (Contd.)

- **Example 1:**

  Show that $f(x) = x^2 + 2x + 1$ is $O(x^2)$.

- **Solution:**

  We can readily estimate the size of $f(x)$ when $x > 1$.

  Because if $x > 1$ then, $x < x^2$ and $1 < x^2$. Thus, if $x > 1$ then,

$$0 \leq (x^2 + 2x + 1) \leq (x^2 + 2x^2 + x^2) = 4x^2.$$

  Consequently, we can take $C = 4$ and $k = 1$ as witnesses to show that $f(x)$ is $O(x^2)$. i.e. $f(x) = (x^2+2x + 1) < 4x^2$ whenever $x > 1$.

  But, can we find another pair of witnesses to this function?

big-O Notation (Contd.)

Let's estimate the size of $f(x)$ when $x > 2$.

When $x > 2$, we have $2x \leq x^2$ and $1 < x^2$ .Thus, if $x > 2$ then,

$$0 \leq (x^2 + 2x + 1) \leq (x^2 + x^2 + x^2) = 3x^2.$$

Consequently, we can take C =3 and $k = 2$ as witnesses to show that $f(x)$ is $O(x^2)$. i.e. $f(x) = (x^2+2x + 1) < 3x^2$ whenever $x > 2$.

big-O Notation (Contd.)

- Note :

  In Example 1, we have two functions,

  - $f(x) = x^2 + 2x + 1$ and
  - $g(x) = x^2$

  Such that "$f(x)$ is $O\big(g(x)\big)$" and "$g(x)$ is $O\big(f\,(x)\big)$"—the latter fact following from the inequality $x^2 \leq x^2 + 2x + 1$, which holds for all nonnegative real numbers x.

  We say that two functions $f(x)$ and $g(x)$ that satisfy both of these $big - O$ relationships are of the ***same order***.

big-O Notation (Contd.)

- Discussion:

When $f(x)$ is $O(g(x))$, and $h(x)$ is a function that has larger absolute values than $g(x)$ does for sufficiently large values of x, it follows that $f(x)$ is $O(h(x))$.

In other words, the function $g(x)$ in the relationship $f(x)$ is $O(g(x))$ can be replaced by a function with larger absolute values.

To see this, note that,

if ($|f(x)| \leq C|g(x)|$ if $x > k$) and ($|h(x)| > |g(x)|$ if $x > k$)

then, ($|f(x)| \leq C|h(x)|$ if $x > k$).

Hence, $f(x)$ is $O(h(x))$.

big-O Notation (Contd.)

- **Example 2:**
  Show that $7x^2$ is $O(x^3)$.

- **Solution :**
  Note when $x > 7$, we have $7x^2 < x^3$ [Multiplying x^2 on both sides of $x > 7$].
  From this we can take $C = 1$ and $k = 7$ as witnesses to establish $7x^2$ is $O(x^3)$.

  Again, when $x > 1$, we have $7x^2 < 7x^3$ [Multiplying $7x^2$ on both sides of $x > 7$].
  From this we can also take $C = 7$ and $k = 1$ as witnesses to establish $7x^2$ is $O(x^3)$.

big-O Notation (Contd.)

x	x^2	7x^2	x^3
0	0	0	0
1	1	7	1
2	4	28	8
3	9	63	27
4	16	112	64
5	25	175	125
6	36	252	216
7	49	343	343
8	64	448	512
9	81	567	729
10	100	700	1000
11	121	847	1331
12	144	1008	1728
13	169	1183	2197
14	196	1372	2744
15	225	1575	3375
16	256	1792	4096
17	289	2023	4913
18	324	2268	5832
19	361	2527	6859
20	400	2800	8000
For $x > 7$,	$7x^2 < x^3$;	$(C, k) = (1, 7)$	

x	x^2	7x^2	7x^3
0	0	0	0
1	1	7	7
2	4	28	56
3	9	63	189
4	16	112	448
5	25	175	875
6	36	252	1512
7	49	343	2401
8	64	448	3584
9	81	567	5103
10	100	700	7000
11	121	847	9317
12	144	1008	12096
13	169	1183	15379
14	196	1372	19208
15	225	1575	23625
16	256	1792	28672
17	289	2023	34391
18	324	2268	40824
19	361	2527	48013
20	400	2800	56000
For $x > 1$,	$7x^2 < 7x^3$;	$(C, k) = (7, 1)$	

big-O Estimates of Functions

- Polynomials can often be used to estimate the growth of functions.
- The leading term of a polynomial dominates its growth by asserting that a polynomial of degree n or $less$ is $O(x^n)$.

- THEOREM 1:

  Let $f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$, where $a_0, a_1, \ldots, a_{n-1}, a_n$ are real numbers.

  Then $f(x)$ is $O(x^n)$.

  i.e. $f(x) \leq Cx^n$, where $C = |a_n| + |a_{n-1}| + \cdots + |a_0|$ whenever $x > 1$. Hence, the witnesses C and $k = 1$ show that $f(x)$ is $O(x^n)$.

big-O Estimates of Functions (Contd.)

- **Example 3:**

  How can big-O notation be used to estimate the sum of the first n positive integers?

- **Solution:**

  Because each of the integers in the sum of the first n positive integers does not exceed n, it follows that

$$f(x) = 1 + 2 + \cdots + n \leq n + n + \cdots n = n^2$$

  This inequality it shows that $f(x)$ is $O(n^2)$, taking witnesses $(C, k) = (1,1)$.

big-O Estimates of Functions (Contd.)

- Example 4:

  Give big-O estimates for the $factorial\ function$ and the $logarithm\ of\ the\ factorial\ function$, where the factorial function $f(n) = n!$ is defined by $n! = 1 \cdot 2 \cdot 3 \cdot \cdots \cdot n$ whenever n is a positive integer, and $0! = 1$.

big-O Estimates of Functions (Contd.)

- Solution:

We can write $f(n) = n!$ in the following way and note that each term in the expression does not exceed n.

$$n! = 1 \cdot 2 \cdot 3 \cdot \cdots \cdot n$$
$$n! \leq n \cdot n \cdot n \cdot \cdots \cdot n$$
$$n! \leq n^n$$

This inequality shows that $f(n)$ is $O(n^n)$, taking witnesses $(C, k) = (1,1)$.

Taking logarithm on both sides we get,

$$\log n! \leq \log n^n$$
$$\log n! \leq n \log n$$

This inequality shows that $\log n!$ is $O(n \log n)$, taking witnesses $(C, k) = (1,1)$.

big-O Estimates of Functions (Contd.)

- As mentioned before, big-O notation is used to estimate the number of operations needed to solve a problem using a specified procedure or algorithm. The functions used in these estimates often include the following:

$$1, \log n, n, n \log n, n^2, 2^n, n!$$

- Each function in this list is smaller than the succeeding function, in the sense that the $ratio$ of $a\ function$ and $the\ succeeding\ function$ tends to $zero$ as n grows without bound.

big-O Estimates of Functions (Contd.)

- Theorem 1 shows that

  If $f(n)$ is a polynomial of degree d, then $f(n)$ is $O(n^d)$. Applying this theorem, we see that if $d > c > 1$, then n^c is $O(n^d)$.

- Taking this into account,
  - If $d > c > 1$, then n^c is $O(n^d)$.
  - If $b > 1$ and $c, d \in \boldsymbol{Z}^+$, then $(\log_b n)^c$ is $O(n^d)$.
  - If $b > 1$ and $d \in \boldsymbol{Z}^+$, then n^d is $O(b^n)$.
  - If $c > b > 1$, then b^n is $O(c^n)$.

Growth of Combination of Functions

- Many algorithms are made up of two or more separate sub-procedures.
- To give a big-O estimate for the number of steps needed, it is necessary to,
  - Find big-O estimates for the number of steps used by each sub-procedure
  - Combine these estimates.
- Big-O estimates of combinations of functions can be provided if care is taken when different big-O estimates are combined. In particular,
  - Estimate the growth of the **sum** of two functions.
  - Estimate the growth of the **product** of two functions.

Growth of Combination of Functions(Contd.)

- Suppose that $f_1(x)$ is $O\big(g_1(x)\big)$ and $f_2(x)$ is $O\big(g_2(x)\big)$.
- From the definition of big-O notation,

$$|f_1(x)| \le C_1|g_1(x)|, \text{when } x > k_1$$
$$|f_2(x)| \le C_2|g_2(x)|, \text{when } x > k_2$$

- Now,

$$\begin{aligned} |(f_1 + f_2)(x)| &= |f_1(x) + f_2(x)| \\ &\le |f_1(x)| + |f_2(x)| \\ &\le C_1|g_1(x)| + C_2|g_2(x)| \\ &\le (C_1 + C_2)|g(x)| \\ |(f_1 + f_2)(x)| &\le C|g(x)| \end{aligned}$$

Where $C = C_1 + C_2$,

$x > k = \max(k_1, k_2)$ and

$g(x) = \max(|g_1(x)|, |g_2(x)|)$

Growth of Combination of Functions(Contd.)

- Suppose that $f_1(x)$ is $O\big(g_1(x)\big)$ and $f_2(x)$ is $O\big(g_2(x)\big)$.
- From the definition of big-O notation,

$$|f_1(x)| \leq C_1|g_1(x)|, \text{when } x > k_1$$
$$|f_2(x)| \leq C_2|g_2(x)|, \text{when } x > k_2$$

- Now,

$$\begin{aligned} |(f_1 f_2)(x)| &= |f_1(x) f_2(x)| \\ &\leq |f_1(x)| \cdot |f_2(x)| \\ &\leq C_1|g_1(x)| \cdot C_2|g_2(x)| \\ &\leq C_1 \cdot C_2|(g_1 g_2)(x)| \\ |(f_1 f_2)(x)| &\leq C|(g_1 g_2)(x)| \end{aligned}$$

Where $C = C_1 \cdot C_2$,

$x > k = \max(k_1, k_2)$

Growth of Combination of Functions(Contd.)

- THEOREM 2:

  Suppose $f_1(x)$ is $O\big(g_1(x)\big)$ and that $f_2(x)$ is $O\big(g_2(x)\big)$. Then $(f_1 + f_2)(x)$ is $O(\max(|g_1(x)|, |g_2(x)|))$.

  - COROLLARY 2.1:

    Suppose $f_1(x)$ and $f_2(x)$ are both $O\big(g(x)\big)$. Then $(f_1 + f_2)(x)$ is also $O\big(g(x)\big)$.

- THEOREM 3:

  Suppose $f_1(x)$ is $O\big(g_1(x)\big)$ and that $f_2(x)$ is $O\big(g_2(x)\big)$. Then $(f_1 f_2)(x)$ is $O\big(g_1(x) g_2(x)\big)$.

Growth of Combination of Functions(Contd.)

- **Example 5:**
  Give a big-O estimate for $f(n) = 3n\log(n!) + (n^2 + 3)\log n$, where n is a positive integer.

- **Solution:**
  Firstly, the product $3n\log(n!)$ will be estimated.
  - $3n$ is $O(n)$
  - From **Example 4** we know that $\log(n!)$ is $O(n\log n)$.
  - Theorem 3 gives the estimate that $3n\log(n!)$ is $O(n^2\log n)$.

  Secondly, the product $(n^2 + 3)\log n$ will be estimated.
  - Because $(n^2 + 3) < 2n^2$ when $n > 2$, it follows that $(n^2 + 3)$ is $O(n^2)$.
  - Thus, from Theorem 3 it follows that $(n^2 + 3)\log n$ is $O(n^2\log n)$.

  Finally, Theorem 2 tells us that $f(x)$ is $O(\max(n^2\log n, n^2\log n))$.

  Using Corollary 2.1 it follows that, $f(n)$ is $O(n^2\log n)$.

Growth of Combination of Functions(Contd.)

- **Example 6:**
  Give a big-O estimate for $f(n) = (x+1)\log(x^2+1) + 3x^2$, where x is a positive integer.

- **Solution:**
  Firstly, a big-O estimate for $(x+1)\log(x^2+1)$ will be found.
  - Note that $(x+1)$ is $O(x)$.
  - Furthermore, $x^2 + 1 \le 2x^2$ when $x > 1$. Hence, if $x > 2$,
  $$\begin{aligned}\log(x^2+1) &\le \log(2x^2) \\ &= \log 2 + \log x^2 \\ &= \log 2 + 2\log x \\ &\le 3\log x\end{aligned}$$
  - This shows that $\log(x^2+1)$ is $O(\log x)$.
  - From Theorem 3 it follows that $(x+1)\log(x^2+1)$ is $O(x \log x)$.

  Secondly, a big-O estimate for $3x^2$ is $O(x^2)$,

  Finally, Theorem 2 tells us that $f(x)$ is $O(\max(x\log x, x^2))$.

  Because $x \log x \le x^2$, for $x > 1$, it follows that $f(x)$ is $O(x^2)$.

Big-Omega and Big Theta

- **big-O** notation is used extensively to describe the growth of functions, but it has limitations.
  - In particular, when $f(x)$ is $O\left(g(x)\right)$, we have an **upper bound**, in terms of $g(x)$, for the size of $f(x)$ for large values of x.
  - However, big-O notation does not provide a **lower bound** for the size of $f(x)$ for large x.
- We use **big-Omega (big-Ω)** notation provide a **lower bound** for the size of $f(x)$ for large x.
- When we want to give both an **upper** and a **lower bound** on the size of a function $f(x)$, relative to a reference function $g(x)$, we use **big-Theta (big-Θ)** notation.

Big-Omega and Big Theta

- Definition 2:
  - Let f and g be functions from the set of $integers$ or the set of $real\,numbers$ to the set of $real\,numbers$. We say that $f(x)$ is $\Omega\big(g(x)\big)$[read as "$f(x)$ is $big - Omega$ of $g(x)$"] if there are constants C and k, such that $|f(x)| \geq C|g(x)|$, whenever $x > k$.
  - There is a strong connection between big-O and big-Omega notation. In particular, $f(x)$ is $\Omega\big(g(x)\big)$ if and only if $g(x)$ is $\mathrm{O}\big(f(x)\big)$

Big-Omega and Big Theta

- Definition 3:
  - Let f and g be functions from the set of $integers$ or the set of $real\,numbers$ to the set of $real\,numbers$. We say that $f(x)$ is $\Theta\big(g(x)\big)$[read as "$f(x)$ is $big - Theta$ of $g(x)$"] if
    - $f(x)$ is $\Omega\big(g(x)\big)$
    - $f(x)$ is $O\big(g(x)\big)$
    - $f(x)$ is of the order $g(x)$ and
    - $f(x)$ and $g(x)$ are of the same order.
    - If there are constants C_1, C_2 and k, such that
      - $C_1|g(x)| \leq |f(x)| \leq C_2|g(x)|$, whenever $x > k$
      - The existence of the constants C_1, C_2 and k tells us that $f(x)$ is $\Omega\big(g(x)\big)$ and that $f(x)$ is $O\big(g(x)\big)$, respectively

Big-Omega and Big Theta

- **Example 7:**
  Show that $3x^2 + 8x\log x$ is $\Theta(x^2)$.

- **Solution:**
  Because $0 \leq 8x\log x \leq 8x^2$, for $x > 1$it follows that,

$$3x^2 + 8x\log x \leq 11x^2$$

Consequently, $3x^2 + 8x\log x$ is $O(x^2)$.
Clearly, x^2 is $O(3x^2 + 8x\log x)$.
Consequently, $3x^2 + 8x\log x$ is $\Theta(x^2)$.

Big-Omega and Big Theta

- **THEOREM 4:**

  Let $f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$, where $a_0, a_1, \ldots, a_{n-1}, a_n$ are real numbers and $a_n \neq 0$.
  Then $f(x)$ is of the order x^n.

- **Example 8:**

  The polynomials,

$$3x^8 + 10x^7 + 221x^2 + 1444,$$
$$x^{19} - 18x^4 - 10112$$
$$-x^{99} + 40001\, x^{98} + 100003x$$

are of orders x^8, x^{19}, and x^{99}, respectively.

THE END