Data Structure and Algorithms

Session-15

Dr. Subhra Rani Patra
SCOPE, VIT Chennai

Time Complexity

(1)
```
for (i = 0; i < n; i++)
{
   statement;
}
```

(2)
```
for (i = n; i > 0; i--)
{
   statement;
}
```

(3)
```
for (i = 1; i < n; i+2)
{
   statement;
}
```

(4)
```
for (i = 0; i < n; i++)
{
   for (j = 0; j < n; j++)
   {
      statement;
   }
}
```

Time Complexity

5

```
for (i=0; i<n; i++)
{
  for (j=0; j<i; j++)
  {
    statement;
  }
}
```

i	j	no. of times
0	0	0
1	0 1	1
2	0 1 2	2
3	0 1 2 3	3
: n		n

$1+2+3+\ldots+n$

$= \frac{n(n+1)}{2}$

$= \frac{n^2+n}{2}$

$O(n^2)$

Time Complexity

6) p = 0

```
for (i=1; p<=n; i++)
{
   p=p+i;
}
```

i	p
1	0
	0+1 = 1
2	1+2 = 3
3	1+2+3
4	1+2+3+4
.	
.	
k	1+2+3+...+k

Assume $p > n$

$$\frac{k(k+1)}{2} > n$$

$$k^2 > n$$

$$k > \sqrt{n}$$

Time Complexity

```
for(i=1; i<n; i=i*2)
{
    statement;
}
```

Assume $i \geq n$

$i = 2^k$

$2^k \geq n$

$k = \log_2 n$

i
1
$1 \times 2 = 2$
$2 \times 2 = 2^2$
$2^2 \times 2 = 2^3$
⋮
2^k

Time Complexity

```
for (i = n, i >= 1, i = i/2)
{
  statement;
}
```

i

n

$\frac{n}{2}$

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

$\vdots$

$\frac{n}{2^k}$

Assume $i < n$

$\frac{n}{2^k} < 1$

Assume $i < 1$

$\frac{n}{2^k} < 1$

$n = 2^k$

$k = \log_2 n$

Time Complexity

9)
```
for (i = 0; i*i<n; i++)
{
  statement;
}
```

10)
```
for (i=0; i<n; i++)
{
  statement;
}
for (j=0; j<n; j++)
{
  statement;
}
```

$i \times i < n$

$i \times i >= n$

$i^2 = n$

$i = \sqrt{n}$

Asymptotic Notations-Definitions

Formal Definition: $f(n) = O(g(n))$ means there are positive constants c and n_0, such that $0 \le f(n) \le cg(n)$ for all $n \ge n_0$.

$1 < \log n < \sqrt{n} < n < n\log n < n^2 < n^3 \ldots < 2^n < 3^n \ldots < n^n$

1

<u>Big - Oh</u>

eg - $f(n) = 2n+3$

$2n + 3 \le 10n \quad n \ge 1$

Asymptotic Notations-Definitions

eg - $f(n) = 2n+3$

$2n+3 \leq 10n \quad n \geq 1$

$\uparrow \qquad \uparrow$

$f(n) \qquad c \quad g(n)$

$f(n) = O(n)$

$2n+3 \leq 7n$

$2n+3 \leq 2n+3n$

$2n+3 \leq 5n \quad n \geq 1$

Asymptotic Notations-Definitions

$1 < \log n < \sqrt{n} < n < n\log n < n^2 < n^3 \cdots < 2^n < 3^n \cdots < n^n$

lower bound   average bound   upper bound

Big - oh

eg - $f(n) = 2n+3$

$\underbrace{2n+3}_{f(n)} \le \underbrace{10}_{c}\, \underbrace{n}_{g(n)} \quad n \ge 1$

$f(n) = O(n)$

$f(n) = O(n^2)$

$f(n) = O(2^n)$

$f(n) \ne O(\log n)$

$2n+3 \le 7n$

$2n+3 \le 2n+3n$

$2n+3 \le 5n \quad n \ge 1$

$2n+3 \le 5n^2 \quad n \ge 1$

$f(n) \le c\, g(n)$

Asymptotic Notations-Definitions

Formal Definition: $f(n) = \Omega(g(n))$ means there are positive constants c and n_0, such that $f(n) >= cg(n)$ for all $n \geq n_0$.

$f(n) = 2n+3$

$2n+3 \geq 1 \times n \quad \forall n \geq 1$

$f(n) \qquad c\,g(n)$

$2n+3 \geq 1 \log n \quad \forall n \geq 1$

$f(n) = \Omega(n)$

$f(n) = \Omega(\log n)$

Asymptotic Notations-Definitions

Formal Definition: $f(n) = \theta(g(n))$ means there are positive constants c_1, c_2 and n_0, such that $c_1 g(n) <= f(n) <= c_2 g(n)$ for all $n \geq n_0$.

$f(n) = \theta(n)$

eg: $f(n) = 2n+3$

$1 \times n < 2n+3 \leq 5 \times n$

$c_1 g(n)$ $f(n)$ $c_2 g(n)$


Thank you