Lab 2

→ Simplified analysis of an algorithm's efficiency.
1) Complexity in terms of input size 'N'.
2) Machine Independent.
3) Basic computer steps.
4) Analyze both time & space.

→ Types of measurement.
1) Worst - case ✓
2) Best - case
3) Average case

} In real practice these are considered too.

→ General Rules
1) Ignore Constants

	cost	Times	
a = 5 x 10;	C1	n	→ O(1)

2) Certain Terms "Dominate" others.
→ O(1) < O(log n) < O(n) < O(n log n) < O(n^2) < O(2^n) < O(n!)
We drop/ignore low-order terms in favor of higher order ones.

→ Constant Time

(1)
a = 5+(11 x 30);
independent of input size 'N'

} c_1 1
O(1)

(2)

x = 11 + 16 ;	c_1	1	
y = 20(15x30);	c_1	1	
cout << x+y << endl;	c_1	1	

} 3 x O(1)
We drop Constants
O(1)

→ Linear Time

(1)

for (int i=0; i≤n ; i++)	c_1	n
cout << i << endl;	c_2	n

} O(N)

(2)

a = (10x5)+6;	c_1	1
for (i=0 ; i≤n; i++)	c_2	n
cout << i << endl;	c_3	n

$T_n = c_1 + nc_2 + c_3 n$
$= n(c_2+c_3)$
$= O(N)$

→ Quadratic Time

```
for(i=0; i<n; i++)              c1   n
    for(j=0; j<n; j++)     }    c2   n.
        cout<<i*j<<endl;   }    c3
```

Example

```
x = (10×5)+6                 }    c1    1
for(i=0; i<n; i++)           }    c2    n
    cout<<i<<endl;
```

$T(n) = c_1 + nc_2$
$= O(N)$

```
for(i=0; i<n; i++)           }    c1    n
    for(j=0; j<n; j++)            c2    n(n)
        cout<<i*j<<endl;
```

$= c_1 n + c_2(n \times n)$
$= c_1 n + c_2 n^2$
$= O(N^2)$

Bubble Sort

First Pass

5	1	4	2	8	→ Swap
1	5	4	2	8	→ Swap
1	4	5	2	8	→ Swap
1	4	2	5	8	

Second Pass

1	4	2	5	8
1	2	4	5	8

⋮

Third Pass

→ Worst Case Time Complexity $O(n^2)$
Best Case " " $O(n)$
Average " " " $O(n^2)$
Space Complexity $O(1)$

```
for(i=0; i<n-1; i++)                      cost   time
{                                          c1     n
    for(j=i+1; j<n; j++)                   c2 = n+n-1+n-2...+1
    {                                         = n(n+1)/2
        if (a[j] < a[i])
        {
            temp = a[i];
            a[i] = a[j];
            a[j] = temp;
        }
    }
}
```

$T(n) = c_1 n + c_2\left(\frac{n(n+1)}{2}\right)$

$= c_1 n + c_2\left(\frac{n^2+n}{2}\right)$

$= c_1 n + \frac{c_2 n^2}{2} + \frac{c_2 n}{2}$

$= \cancel{c_1} n + \cancel{\frac{c_2}{2}}(n^2+n)$

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

$= O(N^2)$

→ How to achieve best case scenario. (3)

a[5]

1	2	3	4	5

```
bool = sorted

do
{
    sorted = true;
    for (int i=0; i<n; i++)
    {
        if (a[i] > a[i+1])
        {
            temp = a[i];
            a[i] = a[i+1];
            a[i+1] = temp;
            sorted = false;
        }
    }
} while (sorted == false);
```

Selection Sort

int a[5] = {18, 10, 7, 20, 2};

Pass 1

18	10	7	20	2
10	18	7	20	2
7	18	10	20	2
2	18	10	20	7

Pass 2

2	10	18	20	7
2	7	18	20	10

Pass 3

2	7	10	20	18

Pass 4

2	7	10	~~18~~ 20	~~20~~ 18
2	7	10	18	20