Time complexity Analysis

*Selection Sort

```
void SelectSort (int ann, int n){
    int select;
    int mindex;
    int minValue;
    int i;

For(select = 0; select <n-1; select++) {
    mindex = select;
    minvalue = ann[select];
    for (i = select + 1; i<n; i++){
        if(ann[i] < minvalue ) {
            minValue = ann[i];
            mindex = i;
        }
    }
    ann [mindex] = ann[select];
    ann[select] = minvalue ;
}
```

	cost	Times
	~~Cases~~	~~Time complexity~~
		Worst Case
For(select = 0; select <n-1; select++) {	c_1	n
mindex = select;	c_2	$n-1$
minvalue = ann[select];	c_3	$n-1$
for (i = select + 1; i<n; i++){	c_4	$(n-1)n$
if(ann[i] < minvalue) {	c_5	$(n-1)(n-1)$
minValue = ann[i];	c_6	$(n-1)(n-1)$
mindex = i;	c_7	$(n-1)(n-1)$
ann [mindex] = ann[select];	c_8	$(n-1)$
ann[select] = minvalue ;	c_9	$(n-1)$

$(n-1)$ one for 1st loop.
$(n-1)$ " " 2nd loop.

The total running time for selection sort is
(worst case)

$$f(n) = c_1 n + c_2(n-1) + c_3(n-1) + \cancel{c_4(n-1)} + c_4 n(n-1) + c_5(n-1)^2 + c_6(n-1)^2 + c_7(n-1)^2 + c_8(n-1) + c_9(n-1)$$

$$= \cancel{c_1 n + (c_2+c_3+c_8+c_9)(n^2-1)(n-1)^2}$$

$$= c_1 n + (c_2+c_3+c_8+c_9)(n-1) + (c_5+c_6+c_7)(n-1)^2 + c_4(n^2-n)$$

Ignore the c_s constants

$$= n + n - 1 + n^2 - 2n + 1 + n^2 - n = 2n^2 - \cancel{2n} n \quad \therefore O(n^2)$$

Best Case

$C_1 \longrightarrow n$

$C_2 \longrightarrow n-1$

$C_3 \longrightarrow n-1$

$C_4 \longrightarrow n(n-1)$

$C_5 \longrightarrow (n-1)(n-1)$

$C_6 \longrightarrow 0$

$C_7 \longrightarrow 0$

$C_8 \longrightarrow (n-1)$

$C_9 \longrightarrow (n-1)$

The total running time for selection sort in best case is

$$T(n) = C_1 n + (C_2 + C_3 + C_8 + C_9)(n-1) + C_4 n(n-1) + C_5 (n-1)^2 + C_6(0) + C_7(0)$$

$$= n + n - 1 + n(n-1) + (n-1)^2$$

$$= 2n^2 - n$$

$$= \Omega(n^2)$$