# Insertion Sort

Inserting element = O(1) - O(n)

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 comp 1 swaps | 4 | 10 | 1 | 6 | 2 | n=6 |
| 2 comps 2 swaps | 4 | 5 | 10 | 1 | 6 | 2 |
| 3 comps 3 swaps | 4 | 5 | 10 | 1 | 6 | 2 |
| 4 comps 4 swaps | 1 | 4 | 5 | 10 | 6 | 2 |
| 5 comps 5 swaps | 1 | 4 | 5 | 6 | 10 | 2 |
| | 1 | 2 | 4 | 5 | 6 | 10 (sorted) |

No. of passes = (n-1) passes

No. of comp. => $1+2+3+4+n-1 : \frac{n(n-1)}{2}$

$O(n^2)$

No. of swaps => $O(n^2)$

✓ Adaptive

| min | max | |
|---|---|---|
| O(n) | $O(n^2)$ | Time |
| O(1) | $O(n^2)$ | swap |
| ↑ sorted | ↑ Descending | |

```
for (i=1 ; i<n ; i++)
{
    temp = a[i];
    j = i-1;

    while (j>-1 && A[j] > temp)
    {
        A[j+1] = A[j]
        j--;
    }
    A[j+1] = temp;
}
```

✓ Stable

# Selection Sort

| | | | | | |
|---|---|---|---|---|---|
| 8 ← i | 2 | 2 | 2 | 2 | 2 |
| 6 | 6 ← i | 3 | 3 | 3 | 3 |
| 3 | 3 | 6 ← i | 4 | 4 | 4 |
| 2 | 8 | 8 | 8 ← i | 5 | 5 |
| 5 | 5 | 5 | 5 | 8 ← i | 6 |
| 4 | 4 | 4 | 6 | 6 | 8 |
| comp = 5 | 4 | 3 | 2 | 1 | |
| swap = 1 | 1 | 1 | 1 | 1 | |

n = 6

no. of comp. $\Rightarrow 1+2+3+4..+(n-1)$

$= \frac{n(n-1)}{2}$ $O(n^2)$

no. of swaps $= n-1$

$O(n)$

```
for (i=0; i<n-1; i++)
{
   for(j=k=i; j<n; j++)
   {
      if (A[j] < A[k])
         k=j;
   }
   swap(A[i], A[k]);
}
```

Adaptive ✗ ↴

same time for already sorted

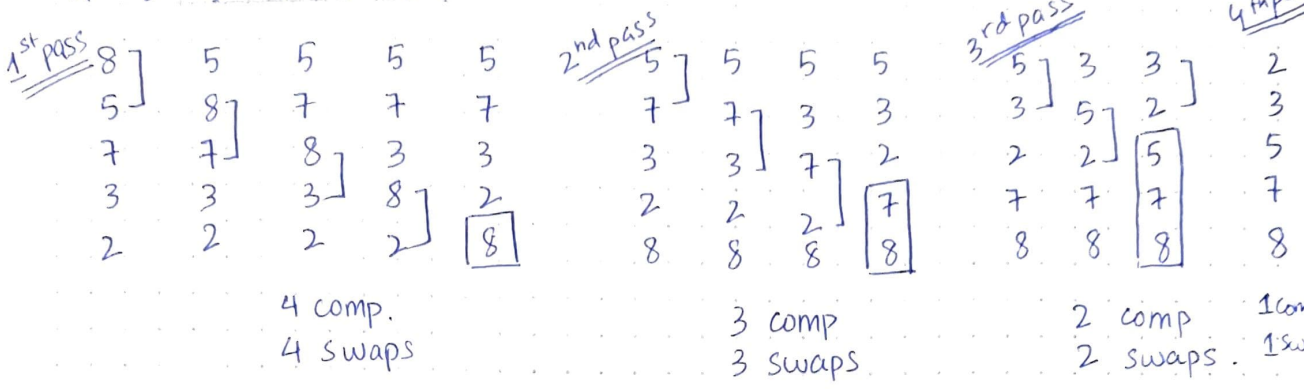
stable ✗ ↴

original order is not preserved.

# Bubble sort

| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| A | 8 | 5 | 7 | 3 | 2 |

n = 5




No. of passes = (n-1)

No. of comparisons => 1+2+3+(n-1) = $\frac{n(n-1)}{2}$ = $O(n^2)$

No. of swaps => 1+2+3+(n-1) = $\frac{n(n-1)}{2}$ = $O(n^2)$

```
for(i=0; i<n-1; i++)
{
   flag=0
   for( j=0; j<n-1-i; j++)
   {
      if (A[j] > A[j+1])
      {
         swap(A[j], A[j+1])
         flag=1;
      }
   }
   if (flag==0) break;
}
```

Adaptive ↗

min = $O(n)$

max = $O(n^2)$

stable ✓

Shell sort

Pass 1

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 23 | 29 | 15 | 19 | 31 | 7 | 9 | 5 | 2 | gap = 4 |
| | ↑i | | | | ↑j | | | | |
| 23 | 7 | 15 | 19 | 31 | 29 | 9 | 5 | 2 | gap = 2 |
| | | ↑i | | | | ↑j | | | |
| 23 | 7 | 9 | 19 | 31 | 29 | 15 | 5 | 2 | |
| | | | ↑i | | | | ↑j | | |
| 23 | 7 | 9 | 5 | 31 | 29 | 15 | 19 | 2 | |
| | | | | ↑i | | | | ↑j | |
| 23 | 7 | 9 | 5 | 2 | 29 | 15 | 19 | 31 | |
| 2 | 7 | 9 | 5 | 23 | 29 | 15 | 19 | 31 | |

```
for ( gap = n/2 ; gap >=1 ; gap/2 )
{
    for ( j = gap ; j< n ; j++)
    {
        for ( i = j-gap; i>=0 ; i-gap )
        {
            if ( a[i+gap] > a[i] )
                break;
            else
                swap ( a[i+gap] , a[i] )
        }
    }
}
```

gap = $n/2$

$O(n^2)$