$\frac{N}{2^k}$ size $= \frac{N}{2^k}$

$\frac{N}{2^k} = 1$

$N = 2^k$

$k = \log N$

So, $O(\log N)$.

Selection Sort

arr[] = | 1 | 7 | 9 | 2 | 3 | 0 |

sorted array => | 0 | 1 | 2 | 3 | 7 | 9 |

what ?

↳ Different ways

↳ Take smallest element and place it in right place.

Date / /

DELTA Pg No.

arr[] → [64 | 25 | 12 | 22 | 11]

j = 0 j++

Round 1:- 64 25 12 22 11

11 25 12 22 64

Round 2:-

11 12 25 22 64

Round 3:-

11 12 22 25 64

Round 4:-

11 12 22 25 64

Total round = (n-1)

Complexity

for i → (n-1)

↳ for j+1 → <n

Space Complexity

↓

O(1)

Time complexity:

a b c d e 4 (n-1)

a b c d e 3 (n-2)

1

1

1

1

1

1

$1+2+3+ \ldots + (n-1)$

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

T.C $\rightarrow O(n^2)$

Best case $\rightarrow$ already sorted $\rightarrow O(n^2)$

worst case $\rightarrow O(n^2)$

Use case: array / list / vector small. then we apply selection sort.

Bubble Sort

arr $\rightarrow \{7, 1, 6, 9\}$

$\downarrow$ in a sorted

$\downarrow$

$\{1, 6, 7, 9\}$

Round 1: 7 1 6 14 9

1 10 7 6 14 9

1 7 10 6 14 9

1 7 6 10 14 9

1 7 6 10 9 14

1 7 6 10 9 | 14 sorted

Total: $(n-1)$ rounds.

Round 2: 1 7 6 10 9 14

~~1 6 7 10~~

1 7 6 10 9 14

1 6 7 10 9 14

1 6 7 10 9 14

1 6 7 9 10 14

Round 3: 1 6 7 9 10 14

1 6 7 9 10 14

1 6 7 9 10 14

1 6 7 9 10 14

1 6 7 9 10 14

1 6 7 9 10 14

sorted

Round 4: like 3 round

=) 1 6 | 7 9 10 14

sorted

Round 5: like 3 round

=) 1 | 6 7 9 10 14

sorted

Worst Case: Round i^{th} => i^{th} largest → right pos

Time Complexity:

1 → (n-1)

2 → (n-2)

1 → 1

(n-1) → 1 comparisons

$1 + 2 + 3 - 1 + \dots + n - 1$

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

(n: size of array)

Space Complexity → O(1)

Best case → already sorted → $O(n)$

worst case → reverse sorted → $O(n^2)$