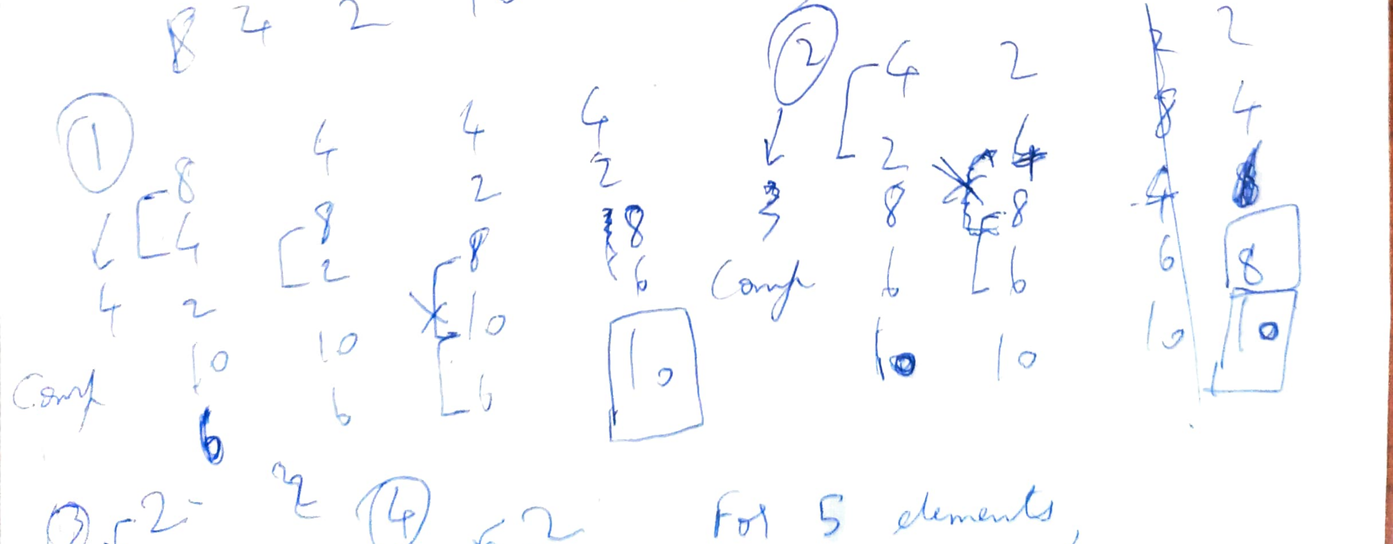
15 Bubble-Sort

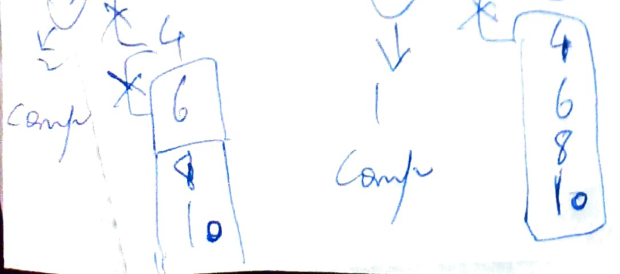
→ Popular Sorting technique based on Brute-Force Technique (Simplest)

→ Brute Force is a straight forward approach to solve a problem. It is usually based on the problem statement and the definitions of the concept involved.

→ In BubbleSort, we compare the adjacent elements of the list and exchange them (if required). By doing it repeatedly, we end up 'bubbling up' the largest element to the last position. In the next iteration, the second largest element bubbles up & so on until, after n-1 iterations, the list is sorted

8 4 2 10 6 (n=5)





For 5 elements,

4+3+2+1 = 10

Algorithm Bubblesort(A[0.....n-1])

// Sorts a given array by bubble sort

// Input: An array A[0...n-1]

// Output: Sorted array A[0.....n-1] in ascending order

for $i \leftarrow 0$ to $n-2$ do // no. of iterations

for $j \leftarrow 0$ to $n-2-i$ do

if (A[j] > A[j+1])

swap (A[j] and A[j+1])

Time complexity

Basic operation → Comparison

Let C(n) denotes the total no. of comparison made for an input size n

$$C(n) = \sum_{i=0}^{n-2} \sum_{j=0}^{n-2-i} 1 = \sum_{i=0}^{n-2} n-2-i+1$$

$$= \sum_{i=0}^{n-2} n-i-1 = (n-1)+(n-2) \cdots 1$$

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

1b) Selection Sort

→ Selection sort is also based on Brute-Force

→ In Selection Sort we scan the entire list to find the smallest element & exchange it with the first element. In the next iteration, we scan the list again from second element to find the smallest among n-1 elements and exchange it with second element and so on until we are left with only one element. ③

→ For n elements, the no. of iterations is n-1.

Eg:-

89 45 68 90 29 34 17 (n = 7)

	1	2	3	4	5	6
89	17	17	17	17	17	17
45	45	29	29	29	29	29
68	68	68	34	34	34	34
90	90	90	90	45	45	45
29	29	45	45	90	68	68
34	34	34	68	68	90	89
17	89	89	89	89	89	90
	6	5	4	3	2	1

Algorithm Selection Sort (A[0 ... n-1])

// Sorts a given array by selection sort

// Input: An array A[0 .. n-1]

// Output: A sorted array A[0 ... n-1]

for i ← 0 to n-2 do // no. of iterations

min ← i

for j ← i+1 to n-1 do

if A[j] < A[min]

min ← j

swap (A[i] and A[min])

Time complexity

Input size = n

Basic operation = Comparison

Let $C(n)$ denotes the Total no. of comparisons

$$C(n) = \sum_{i=0}^{n-2} \sum_{j=i+1}^{n-1} 1$$

$$= \sum_{i=0}^{n-2} n-1-i \cancel{-1+1}$$

$$= \sum_{i=0}^{n-2} n-1-i = \frac{n(n-1)}{2} = \frac{n^2-n}{2} = n^2$$