DSA
INTERNSHALA
Searching and Sorting

Searching

Linear Search O(n)

1. Set Count = 1
2. Repeat steps 3,4 while count <= 10
3. If val = arr[count], then:
   Write: value found at location number, count
   Return
   [end of if]
4. Set count = count + 1
5. Write: no such value exists
6. return

Binary Search $O(\log_2 n)$

Applicable only on sorted arrays

1. set lft = 1 , rt = n , md = (lft+rt)/2
2. repeat steps 3,4 while lft <= rt
3. if val>ar[md], then
   set lft = md + 1
   else
   if val<ar[md], then
   set rt = md – 1
   else
   write: found at location number, md
   return
   [end of if]
4. set md = (lft+rt)/2 [updating the value of md]
5. write: no such value exists
6. return

Bubble Sort

1. repeat for I = 1 to n-1 [outer loop for pass]
2. repeat for j = 1 to n-I [inner loop for comparisons]
3. if arr[j] > arr[j+1]
   swap : arr[j] and arr[j+1]
   [end of if]
   [end of j loop]
   [end of I loop]
4. return

eg : 5 3 4 2 1 3 2

pass 1: 3 4 2 1 3 2 5

pass 2: 3 2 1 3 2 4 5

pass 3: 2 1 3 2 3 4 5

pass 4: 1 2 2 3 3 4 5

it will then do 3 more pass with no change. We can set a check variable which will check if a swap is made or not. If in any pass there has been no swaps then we can break out of the loop.

Selection Sort

N-1 passes takes place

In each successive pass number of comparisons keep decreasing by one

1. repeat for I: 1 to n-1 [outer loop for passes]
2. repeat for j = i+1 to n [inner loop for comparisons]
3. if arr[i] > arr[j], then
   swap(arr[i],arr[j])
   [end of if]
   [end of j loop]
   [end of I loop]
4. return