

3) Bullete SC Dorst Best Avg. (1)  $O(n^2)$  $O(n^2)$ Bubble (h) 0(1) O(n2)  $O(n^2)$  $O(n^2)$ Selection 0(1) 0(62)  $O(n^2)$ Insertion O(n)(n) O(nlogn) O (nlgn) O(nlogn) O(bogn) to O(n) Merge  $O(n^2)$ O (n log n) O (n log n) Chuich O(nlogn) O (n logn) O (n logn) Heap 0(n+k) -O(n+k)Counting O(n+b) 0(n+b)\*
log b (k) Radix Online algorithm
1. insertion a insertion 5. redix.

S.) Recursive Binary Search —
function bineary Search (avoir, low, high, target)

Let high > low. if our [mid] == tweet seturn binary search (ase, low, mid-1, target) elseif are fried > target seturn binary Search (aver, mid + 1, high, target) seturn-1 Storative Binary Search function binary hearch (099, target) Moro = 0 high = length of our - 1. while (low <= high)

mid = low + (high - low) /2.

if (arr [mid] == target)

return mid else if (arr [mid] < target) high = mid -1; Lineway Search Binary Seach TC = O (log n)  $SC = O(\log n)$ 

6.) Recurrence relation -T(n) = T(n/2) + O(1)T(n) = time taken to search within array of time taken to search within half half 7)# include (iostriam) using namespace std; void findIndexes (int ACI, int size, int K) for (int i=0; ix size -1; ++1) for (int j = i+1; j < xige; ++j) if (A[i] + A[j] == K)cout << "Indexes" << i< "," << j << ind)

cout << "Numbers" << A[i] << "," < A[j] << endl cout<< "No such pair exists" « und);

8.) Insertion not oit is stable & in-place, making it suitable for memory usage is a concern. Selection nort is more imp. Than performance. 9. In this context of arrays, an invorsion seems to a pair of elements over [i], are [i] such that i j, but (are [i] > are [j]). In simple turns, it indicates that the elements are out of order relative # include < ipotream > long long (int age [], int timp [], int left, int mid, int
eight) int i = mld
int k = left long long invorsions = 0. while is mid It i = eight) if are [i] <= ara [

while (1 < mid) arr j++ seturn enversions; long long murgesort (int was [], int temp[], int left, int eight long long inversions = 0; ager, temp. ectura inversions;

10.) The best case complicity (TC) occurs when the partition process always picks the middle element as the pivot. TC = O(n legn) The point case time complexity happens when the pivot selection consistently results in highly unbalanced partitions, such as when the smallest or largest element is always chosen as the pivot. chosen as the pivot.  $TC = O(n^2)$ . 1) Murge Soit -Best Case - T(n) = dT(n/2) + O(n)Norst (ase - T(n) = 2T(n/2) + O(n) Quick Sot -Best Case - T(n) = 2T(n/2) + O(n) Worst Case \_ T(n) = T(n-1) + O(n) · both have best TC = O (n logn)
word TC = O (n logn) · both use divide & conquer technique.

· both are comparison—based stating algorithms. 1. menge soit is stable soit, whereas Quick stat is not .
2. Murge soit requires more space, whereas Quick soit is inplace sorting algorithms

process so that it igual elemento unhocessarily. void stablelection (int par [], int n) for (int i=0; icn-1; ++i) int min\_index = int j= i+1; j<n; ++j) our [] < our [min\_index min\_index = j while (min-index > i sor min index -1 min [i] = min value;

13) # include <iootram > void bubblesort (int are [], int n) for (int i=0; kn; i++) book swapped = false;