O(1) = algorithm that runs for the same ammount of time, no matter how big the ‘array’ is

O(N) = правопропорционален алгоритъм на броя данни

O(N^2), O(N^3)… = as the number of items increases the number of steps increases many more times

O(log N) = за подадени N елемента на всяка стъпка броя операции ще намаля наполовина

O(n \* log n) = O( log(n!) ) = O( log(n) + log(n-1) + log(n-2) + … + log(1) )

Dynamic array:

Insert – O(n)

Insert at end – O(1) amortized

Remove – O(n)

Access – O(1)

Search – O(n)

Can be used in Stack, Queue, Priority Queue, Trees, Graph

Linked List:

Insert – O(n)

Insert at Front – O(1)

Insert at End – O(n) if no “tail” pointer is stored

Remove – O(n)

Access – O(n)

Search – O(n)

\*For Circular list: insertion at front/end is O(1) because last inserted element points to front and we keep it as a pointer

Can be used in Stack, Queue, Priority Queue, Trees, Graph

Stack:

Push – O(1)

Pop – O(1)

Top – O(1)

Search – O(n)

Свързано и последователно представяне.

Приложения: Expression evaluation and syntax parsing, Backtracking, Call stack, For representing Recursion as iterative – DFS, Topological Sorting

Queue:

Push – O(1)

Pop – O(1)

Front – O(1)

Search – O(n)

Uses:

When a resource is shared among multiple consumers.Examples include CPU scheduling, Disk Scheduling.

When data is transferred asynchronously (data not necessarily received at same rate as sent) between two processes. Examples include IO Buffers, pipes, file IO, etc.

In algorithms: BFS

Binary Search Tree:

Insert – Best case: O(log(n)) , Worst case: O(n)

Remove – Best case: O(log(n)) , Worst case: O(n)

Search – Best case: O(log(n)) , Worst case: O(n)

Worst case is because if you insert 1, 2, 3 ,4, 5….1000 it will become a linked list

Binary Tree:

Search – O(n)

Remove – O(n)

// No exact insertion definition

AVL Tree, Red Black Tree:

Insert – O(log(n))

Remove – O(log(n))

Search – O(log(n))

Graph:

Create Vertex: O(V)

Create Edge: O(V)

Get Edge: O(V)

Get Vertex: O(V)

BFS, DFS:O(V+E)

Topological: O(V+E)