Listas enlazadas

1

.

Inserción:

–

Al inicio: (O(1))

–

Al ﬁnal: (O(n)) si no se mantiene un puntero al último nodo, (O(1)) si se tiene un

puntero.

–

En el medio: (O(n)).

2

.

Borrado:

–

Al inicio: (O(1)).

–

Al ﬁnal o en el medio: (O(n)).

3

.

Búsqueda: (O(n)).

Pilas (Stacks)

1

.

Push (inserción): (O(1)).

Pop (eliminación): (O(1)).

Peek (acceso al tope): (O(1)).

2

.

3

.

Colas (Queues)

1

.

Enqueue (inserción al ﬁnal): (O(1)).

Dequeue (eliminación del inicio): (O(1)).

Peek (acceso al frente): (O(1)).

2

.

3

.

Colas de prioridad (Heaps)

1

.

Inserción: (O(\log n)).

2

.

Extracción del máximo/mínimo: (O(\log n)).

Acceso al máximo/mínimo: (O(1)).

3

.

Arreglos (Arrays)

1

.

Acceso: (O(1)).

2

.

Inserción o eliminación:



–

–

–

Inicio: (O(n)).

Medio: (O(n)).

Final: (O(1)) si hay espacio.

Árboles Binarios de Búsqueda (BST)

1

.

Búsqueda:

–

Mejor caso: (O(\log n)).

–

Peor caso (desbalanceado): (O(n)).

2

.

Inserción:

–

Mejor caso: (O(\log n)).

–

Peor caso: (O(n)).

3

.

Eliminación:

–

Mejor caso: (O(\log n)).

–

Peor caso: (O(n)).

Árboles Balanceados (AVL, Red-Black Trees)

1

.

Búsqueda, inserción y eliminación: (O(\log n)) en todos los casos.

Hash Tables

1

.

Inserción: (O(1)) (promedio), (O(n)) (peor caso si hay muchas colisiones).

2

.

Búsqueda: (O(1)) (promedio), (O(n)) (peor caso).

3

.

Eliminación: (O(1)) (promedio), (O(n)) (peor caso).

Grafos (representación por listas de adyacencia o matriz de

adyacencia)

1

.

Búsqueda de vecinos:

–

Lista de adyacencia: (O(k)), donde (k) es el grado del nodo.

–

Matriz de adyacencia: (O(n)).

2

.

Inserción de aristas:



–

–

Lista de adyacencia: (O(1)).

Matriz de adyacencia: (O(1)).

3

.

Eliminación de aristas: Similar a la inserción.