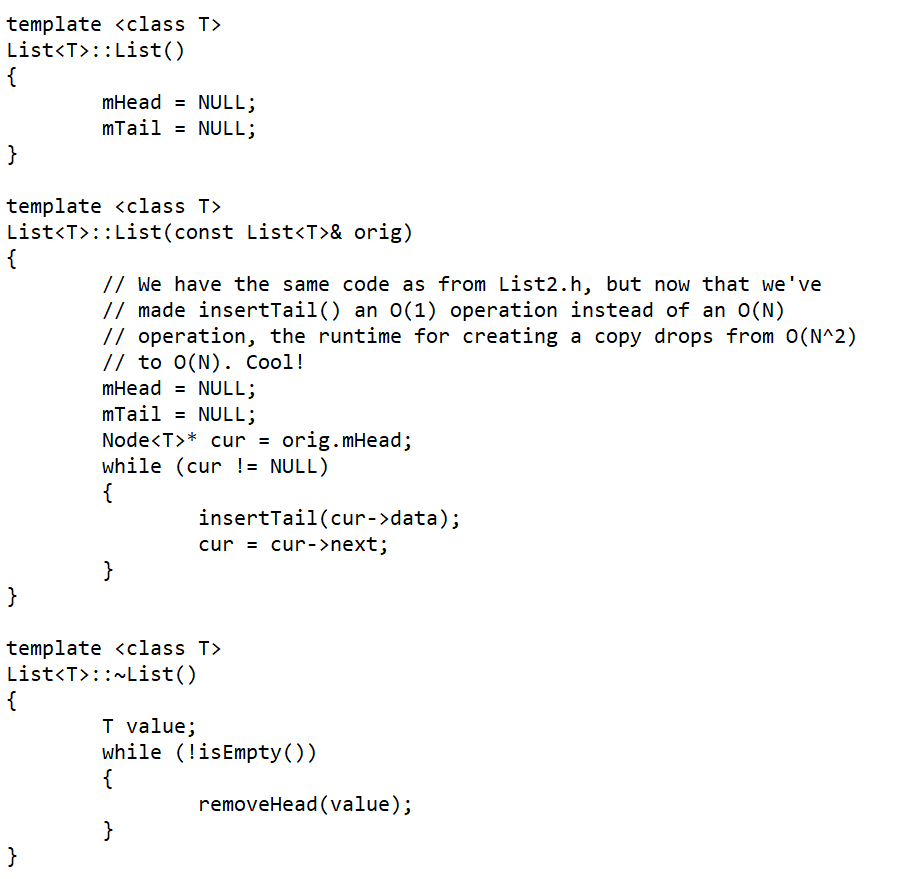
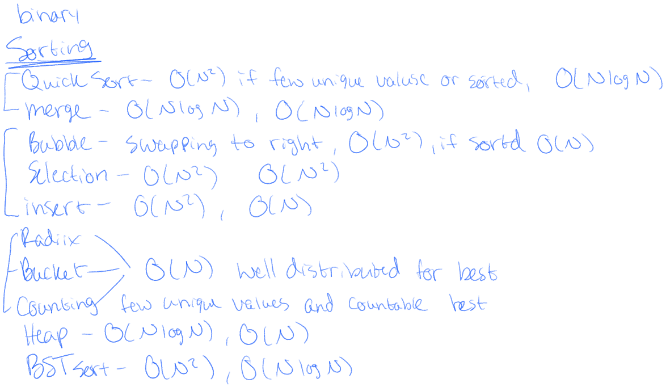
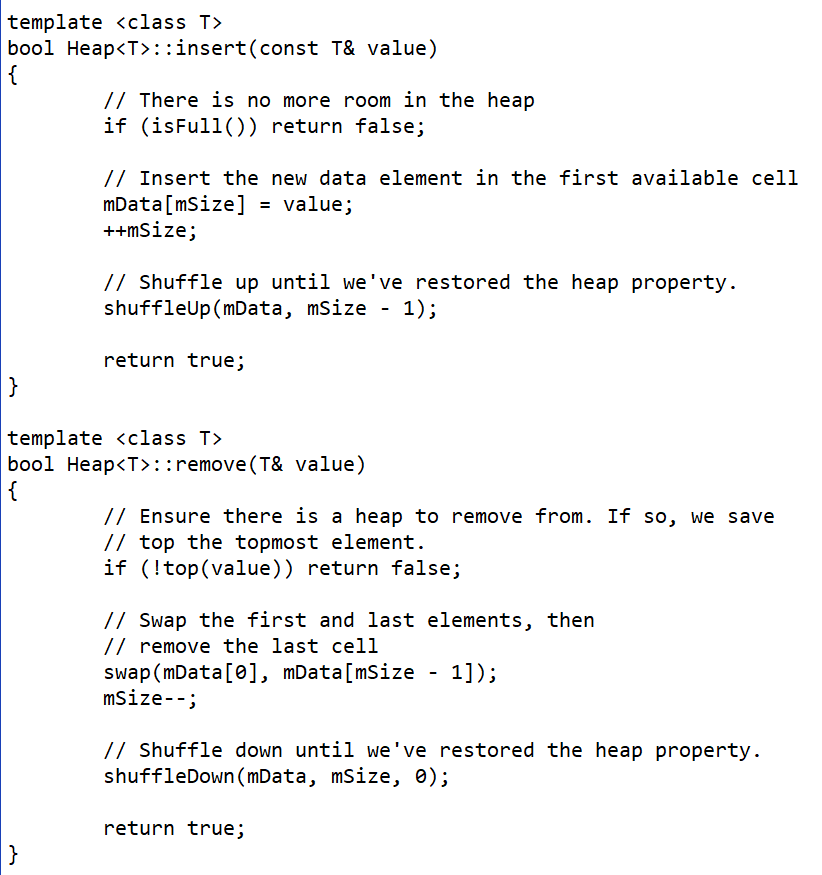
A screenshot of a cell phone screen with text

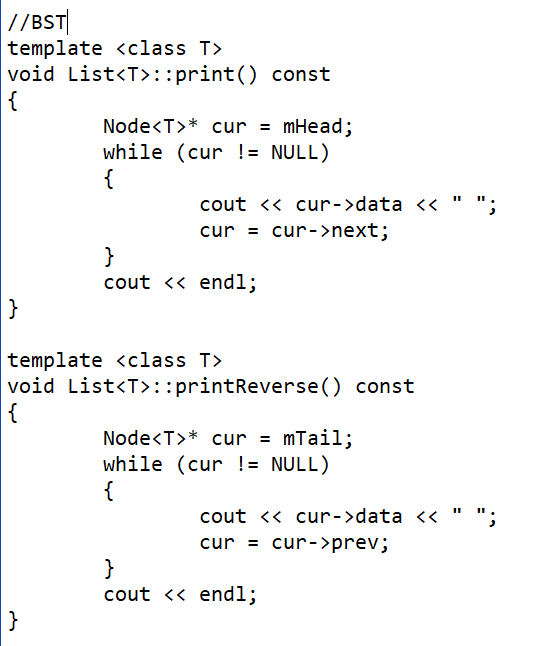
Description generated with very high confidence



A screenshot of a cell phone

Description generated with very high confidence





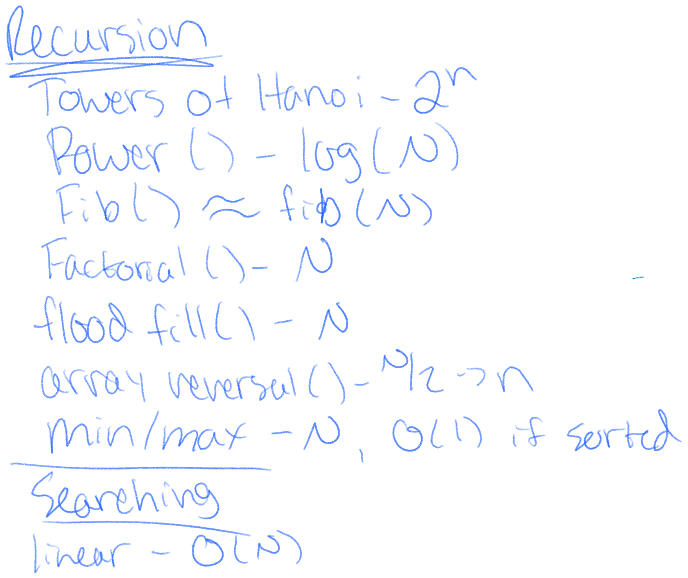
**Vector/Array: insert** – O(1) at end, O(N) at middle, **remove** – O(N), **search** – O(N) unsorted, O(logN) sorted

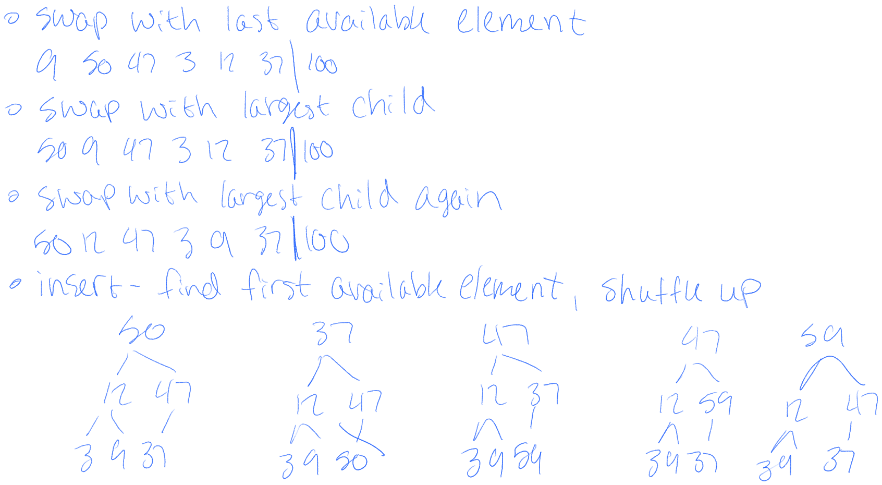
**Linked List: insert** – O(N), **remove** – O(N), **search** – O(N)

**BST: insert** – O(logN), **remove** – O(logN), **search** – O(logN)

**Hash Table: insert** – O(1), **remove** – O(1), **search** – O(1) separate chaining is with Linked List

Heaps are complete and balanced, have the heap property, and max value is on top, can only remove head

2*i* + 1 for left, 2*i* + 2 for right



**BST** is like Linked List but with 2 ->next

Remove head by swapping with successor then reroute with leaf then delete

If balanced insertion is O(logN), otherwise its O(N)

**Queue** – remove head, insert tail or remove tail, insert head

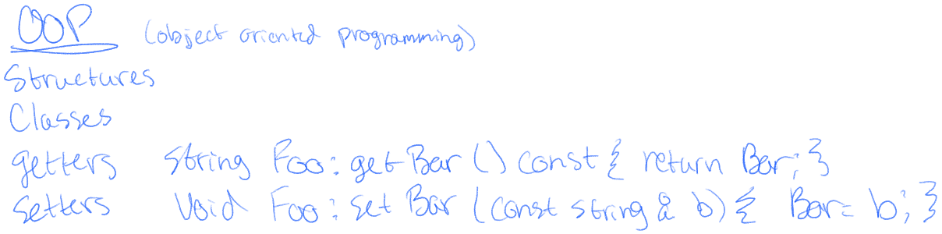
First in, First Out (think of shopping line)

**Stacks** can only access top

Pop to remove top, push to add to top, top to view top but cant change

No binary search for **Linked List**

removeHead() – move head pointer and delete node

**Vectors** – resizable arrays