Pointers

* Types of pointers →

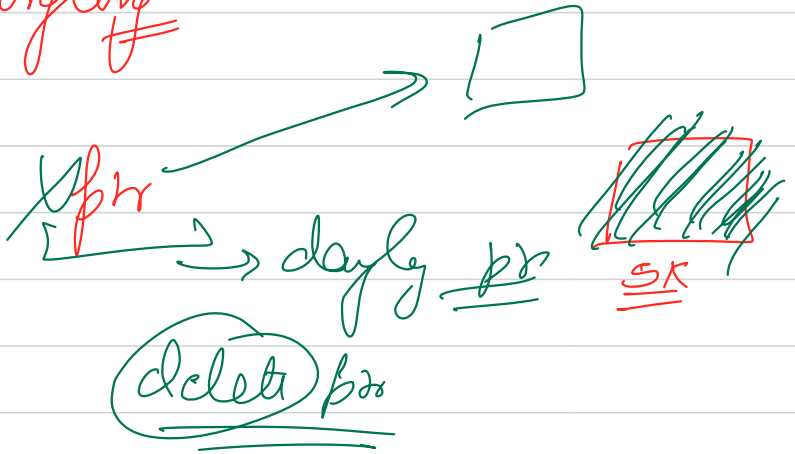
↳ Dangling Pointer

↳ Void Pointer

↳ Null pointer

↳ Wild pointer

* dangling pointers → You initialized any pointer variable and then freed up the memory then that ptr becomes dangling.

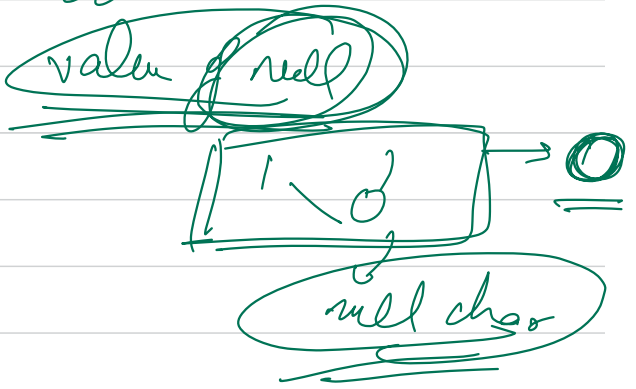


* Wild pointer → A pointer that has been declared but not initialized

```
int *ptr;
```

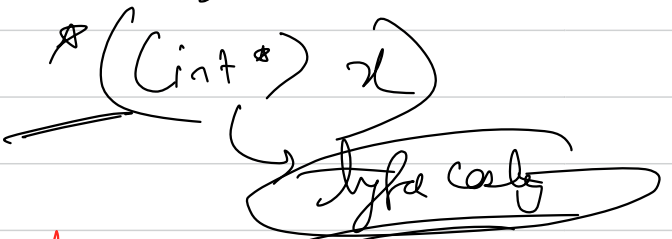
* Null pointer → It is a special type of pointer which points to null or no where. If we don't know address to assign, assign as NULL.

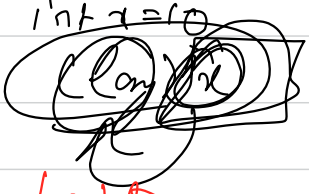
int *p = NULL;



Void pointer → special ptr → void * → this * is a pointer that points to some memory area, which isn't have specific type

↳ Void pointer cannot be dereferenced
↳ So we first typecast & then deregen





↳ we can't do any arithmetic on void pointer

Difference btw pointers & references

↳ pointers can be declared first & initialized late. But references are always initialized during declaration

↳ pointers can be reassigned. No reassignment for references

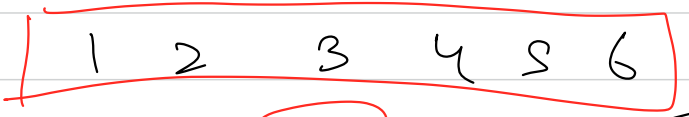
↳ pointer has a variable & it's own bucket in memory where it stores the address.

↳ References have same bucket as no of variable

↳ pointers can be NULL but references can't

↳ pointers support arithmetic while references not

↳ Q) You have a sorted array. & a variable S. You need to find a triplet that sums upto S.



S=10 → Yes

(6, 3, 1)

$N \leq 10^7$

Sol: For any a[i] → find a pair from [i+1, n-1] such that S - a[i] == sum of pair

Q: Given 2 sorted arrays, merge them into one single sorted array

[1, 12, 15] → N [4, 6, 7] → m

↳ [1, 2, 4, 5, 6, 7]

$N, m \leq 10^6$

O(N+m)

Merge 2 sorted arrays

↳ Subroutine in mergesort

sorted

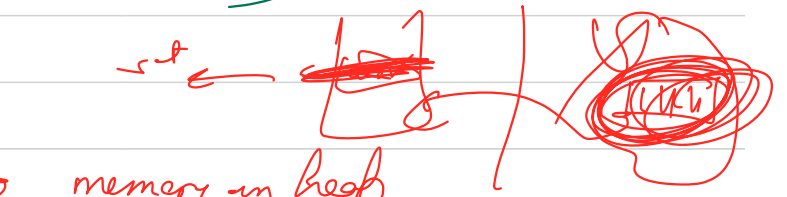
[1, 2, 5]$_3$

2^5 sorted

[4, 6, 9]$_3$

[1, 2, 4, 5, 6, 9]$_6$

$m + n$



memory in heap

int *x = new int;

memory for array with

int *arr = new int[10];

Q: sorted array & a value D. find all the pairs with difference equal to D.

1, 2, 3, 4, 5 D = 3

(2) (4,1) (5,2)

2^{10} 2

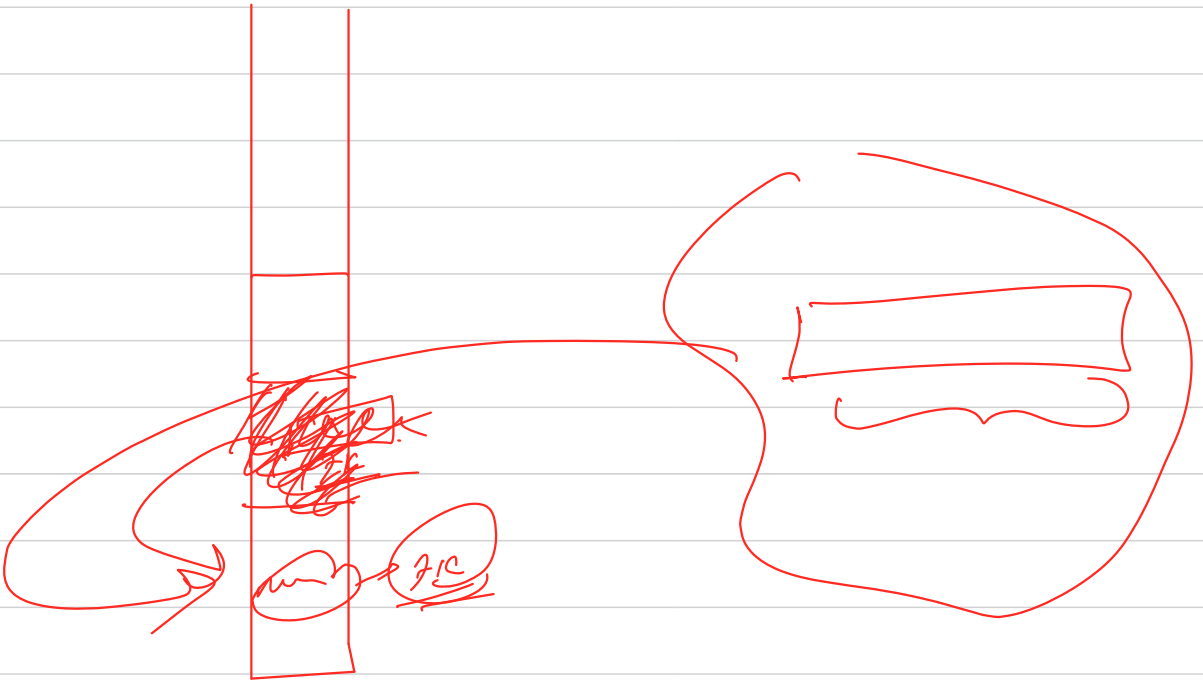
1, 2, 3, 4, 5

D=3

1<D

2<3

Diff decn by either dec right for or inc left for



↳ pointers can be deeply nested. But not multiple references.