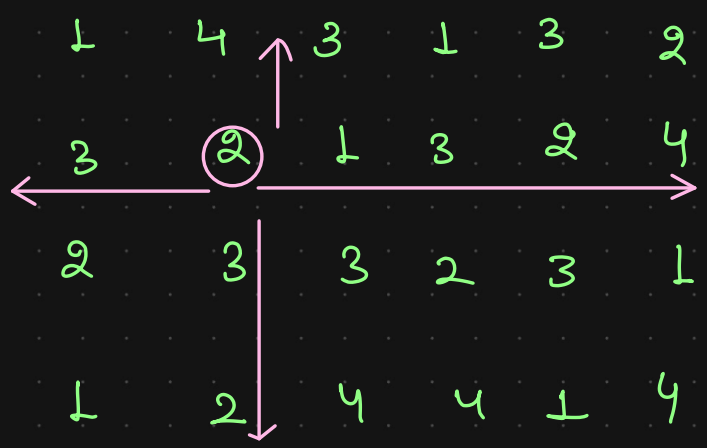
Trapping Rain water - 2

arr → [[1,4,3,1,3,2],[3,2,1,3,2,4],[2,3,3,2,3,1]]

length → size of arr[0]

width → arr.size

height → arr[i][j]



Trapping Rain water - 1

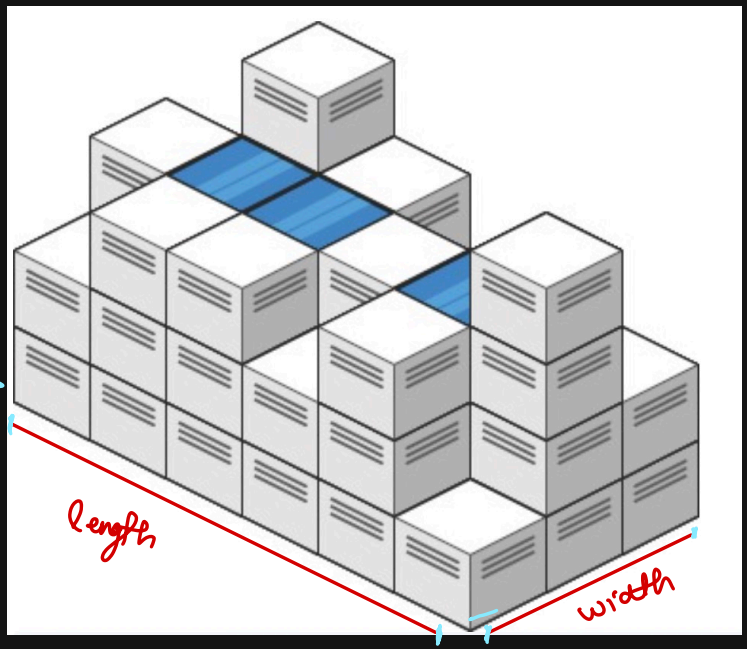
1-D problem

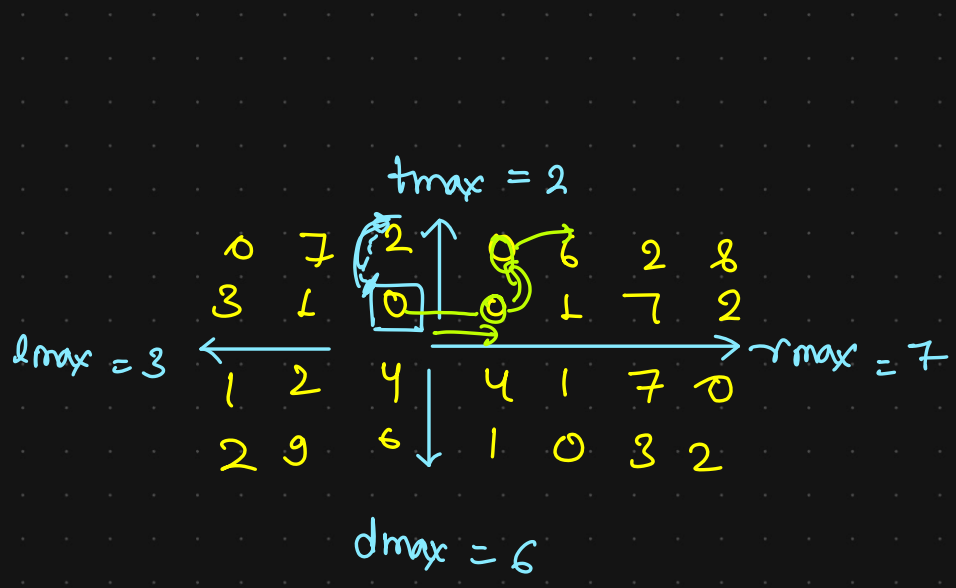
find max from Both direc.

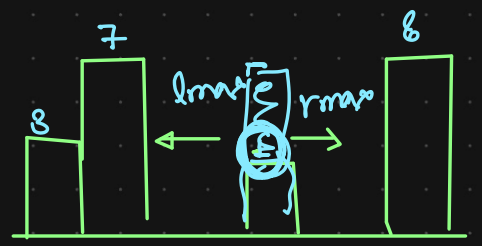
min. max value & find water

Trapping Rain water - 2

- Basic calculator - 1
- Basic Calculator - 3
- No. of valid Subarray
- lexicographically Smallest subseq







lmax = 7
rmax = 8

self value = 1

selfval- > min(lmax, rmax)

No water

Min = (lmax, rmax, tmax, dmax)

= 2

waterer = min - ht[i][j]

~~Not functional~~ (lmax, rmax, tmax, dmax)

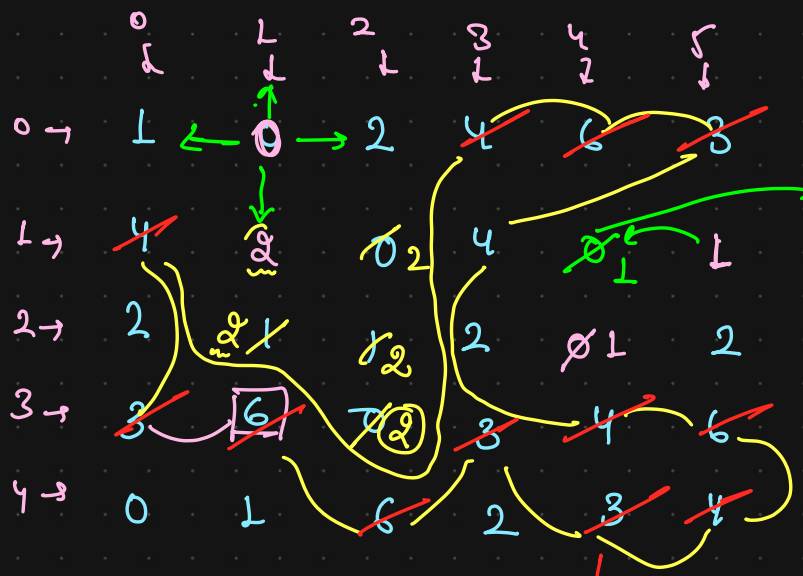
Important factor : Boundary of building

water += min(lmax, rmax) - selfval

Height of Building :→

Min from Boundary Element :

→ priority Queue



0-1 →

left is already visited marker

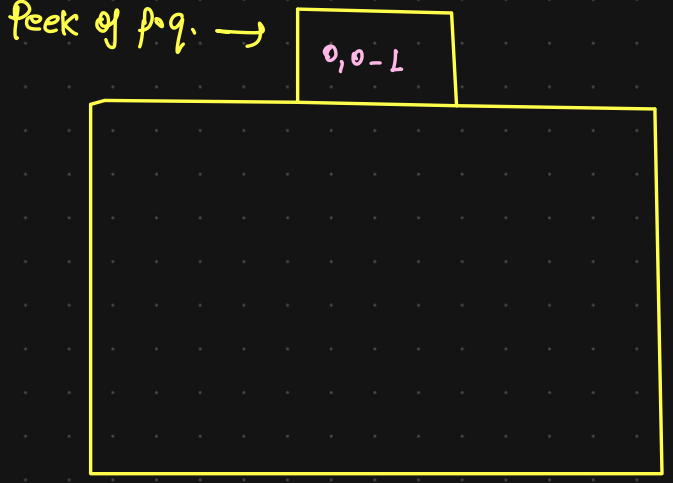
added right is already added.

To create impact of ↓ (i.e. ngbr nbs)

We add '0' as ↓ in pq.

Boolean visited

All directions are eligible added or solved



water += [current ht for removal of pq − ht of additional building]

water = 0 + 1 + 1 + 2 + 1 + 1 + 2 = 8 unit

* Boundary Shrinkage
* visited
* water add if nbs are smaller
* to make impact increase ht of nbr if they are smaller

steps :

① Add boundary element in priority Queue. [i.e. top wall, left wall, down wall, right wall]

② while pq.size() > 0

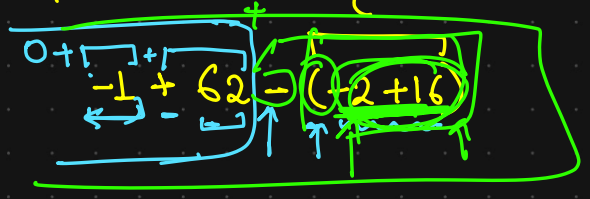
→ Remove peek element and add its nbrs in same condition, ie. nbr must be a valid index and unvisited.

→ if nbr is small then add contribution in water and make impact of current ht on nbr.

→ if nbr is higher add as it is.

Basic Calculator-1 ⟶

Example →



1 + 62

operator allowed → + , -

opr1 opr2

sign = +1 * -1 = -1

+ + → +ve
+ - → -ve
- + → -ve
- - → +ve

priority of + and - is same, depends on which operator is encounter first.

-1 + 62

sum = 0 + -1 + 62 = 0 + 61 = +ve 61 -ve -1 + 61

sign = +1 * +1 * -1 = +1 * +1

val = val * sign 1, 62 * 16

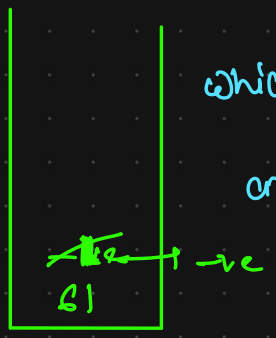
= 1 62 * 16

a + b + (c + d)

split in numbers

addition

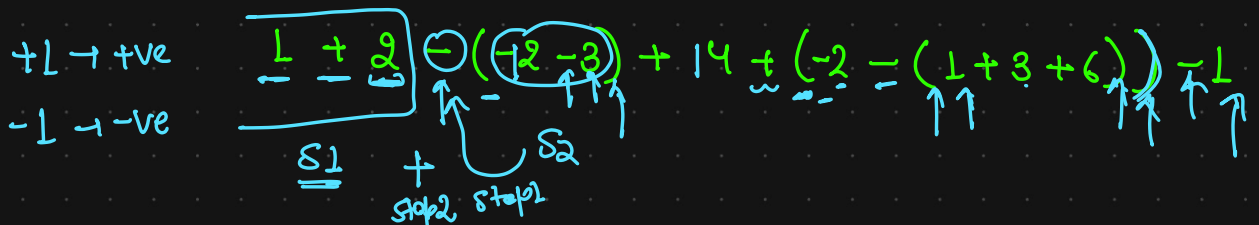
which operator we are solving = +ve



which are +ve or -ve

+1 → +ve

-1 → -ve



$-10 + (-2)$

$1 + 2[-(-12) + (3)] + 14$

$-12 * 1 = 12 + 32 = 20 + 1$

$= 19$

Answer

Sum = ~~0 1 2 0 -12 -15 * 1 = 15 + 32 = 18 32 0 12 0 1 1 10 -12 -20~~

Sign = ~~+1 -1 +1 -1 +1 -1 +1 -1 +1 -1 +1 -1~~

Val = ~~1 2 12 3 14 2 1 3 5 1~~

Val * Sign = ~~1 2 -12 -3 14 -2 1 3 5 -1~~

sum = sum + val * sign

goal → a + b + c + d

Addition operator. Reset sign

val → complete number. ~~further process~~

+ve → nothing to do

-ve → multiply sign with "-1".