Plan - Sunday

Saturday, 18 September 2021 10:31 AM

Morning

1. Min Domino Rotation for Equal Rows
2. Gas Station / Petrol pump
3. Sum of Subseq. width
4. Smallest Range from k-list
5. Car pooling

Evening

1. Max. product Subarray
2. Min size Subarray sum greater than equal to target
3. Max. average Subarray.
4. Min length of String after removing similar end.
5. Find all duplicate.

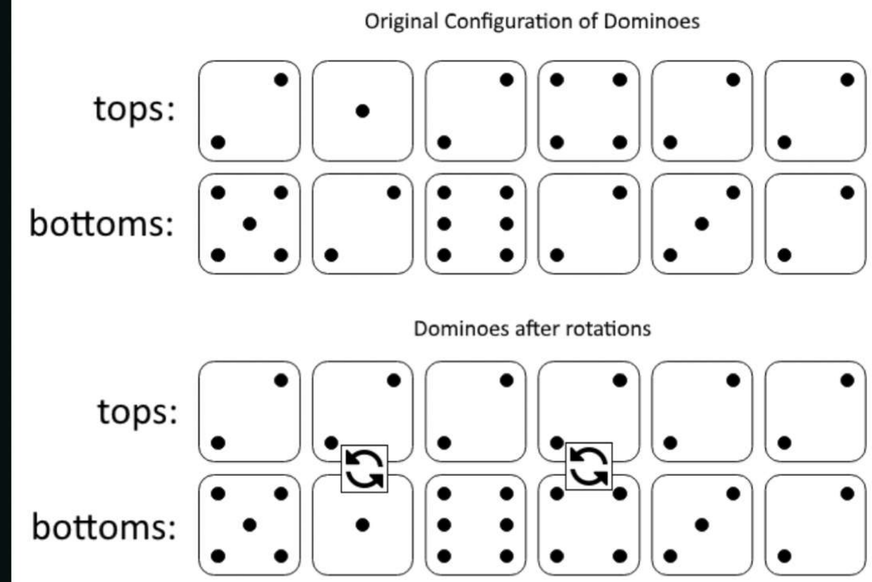
* 6. Min height → GFG

Min Domino Rotations for Equal Rows

Saturday, 18 September 2021 10:31 AM

Min Rotation, if possible.

Example 1:




```
Input: tops = [2,1,2,4,2,2], bottoms = [5,2,6,2,3,2]
Output: 2
```

Min = Result

top as → 2 → count1 → Rotation to make top as 2

top as → 5 → count2 → Rotation to make top as 5

Bottom as → 5 → count3 → Rotation to make Bottom as 5

Bottom as → 2 → count4 → Rotation to make Bottom as 2

Count 1 → Count1 = ∅ ↙ 2

Count3 = ∅ ∞

top: 2 1 2 4 2 2

Bottom: 5 2 6 2 3 2

count4 = ∅ ↙ 2 ↙ 3

count1 = 2
count2 = ∞
count3 = ∞
count4 = 3

Min → we can solve it

Count2 = ∅ ✗ ∞ ← marker

we can identify in single iteration.

```
——— loop ———
if( count1 != ∞){
    if( top[i] == val1 ){
      // Nothing to do
    }else if( Bottom[i] == val1 ){
      count1++;
    } else {
      count1 = Integer.MAX_VALUE;
    }
}      Same 'if' for all counts.
```

count1 =
count2 =
count3 =
count4 =

val1 =
val2 =

Follow same code for all counts.

Gas Station

Saturday, 18 September 2021 10:31 AM

Initial quantity in fuel tank = 0

Input: gas = [1,2,3,4,5], cost = [3,4,5,1,2]

Output: 3

Explanation:

Start at station 3 (index 3) and fill up with 4 unit of gas. Your tank = 0 + 4 = 4

Travel to station 4. Your tank = 4 - 1 + 5 = 8

Travel to station 0. Your tank = 8 - 2 + 1 = 7

Travel to station 1. Your tank = 7 - 3 + 2 = 6

Travel to station 2. Your tank = 6 - 4 + 3 = 5

Travel to station 3. The cost is 5. Your gas is just enough to travel back to station 3.

Therefore, return 3 as the starting index.

gas available at current fuel pump

distance of next fuel pump

mileage = 1 unit/km

capacity of fuel tank is infinite

To find → find a starting point from which we can start and after a complete circle, we will end at that point.

NOTE: ① If there exist a solution then it is guaranteed unique.

② Both are input array are non Empty

③ Each elenet in input array are non -ve

Fact → ① $\sum gas - \sum cost < 0$ → Not possible to make a circular path → return -1

② $\sum gas - \sum cost \geq 0$ → there must be a path, $\sum gas \geq \sum cost$ → solve

we have to solve only if

vehicle

Mileage = 1 km / 1 Ltr

gas available → (Ltr)

P1	P2	P3	P4	P5
1	2	3	4	5
0	1	2	3	4

distance of next gas station/petrol pump (km) →

P1	P2	P3	P4	P5
3	4	5	1	2
0	1	2	3	4

Example 2:-

5	1	2	3	4
4	4	1	5	1

o/p → 3rd index

fuel: 0 4 2 8 6 7 4 6 8 5 0

What we have to find??

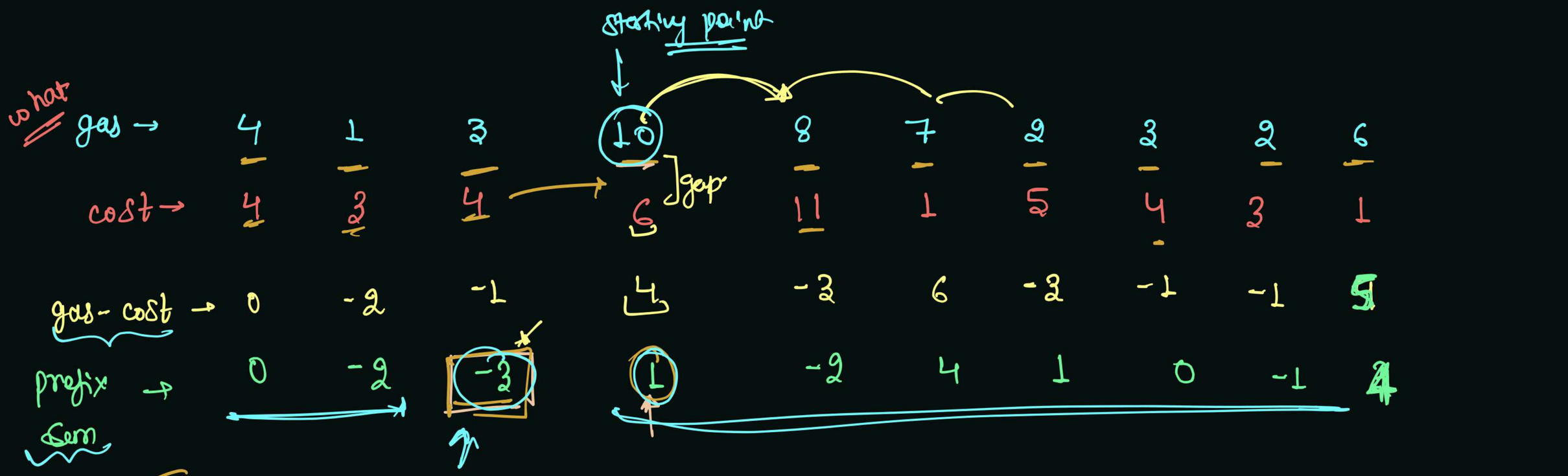
→ initial point from which we can start and make a complete circle to it

$\sum gas - \sum cost < 0$ → Circular traversal is not possible, return -1.

$\sum gas - \sum cost \geq 0$ → It must have a circular path.

$\sum$ → Summation

Total Sum



Find most -ve prefix sum index and start with next index of it, than we can complete our cycle.

fuel = ∅ 10 4 12 1 8 7 8 4 7 3 5 2 8 7 11 7 8 5 8 (4)

Let's Analyse why it works??

→ we have information that i^{th} prefix sum is very smallest.

To prove → $gas[i+1] - cost[i+1] \geq 0$

proof → $prefix[i+1] \geq prefix[i]$ ← prefix[i] → least value

$prefix[i] + gas[i+1] - cost[i+1] \geq prefix[i]$

$\Rightarrow$ $gas[i+1] - cost[i+1] \geq 0$ → gap is available to move at next gas station

Now from above 'i+1' can we reach i+2, check if it is possible move?

To prove → $[gas[i+1] - cost[i+1]] + gas[i+2] - cost[i+2] \geq 0$

gap gap

buffer $gas[i+1] - cost[i+1] \geq 0$

from previous proof.

$\Rightarrow prefix[i+2] \geq prefix[i]$ (least -ve)

$= prefix[i+1] + gas[i+2] - cost[i+2] \geq prefix[i]$

$= prefix[i] + gas[i+1] - cost[i+1] + gas[i+2] - cost[i+2] \geq prefix[i]$

$= gas[i+1] - cost[i+1] + gas[i+2] - cost[i+2] \geq 0$

proved

To prove → $gas[i+1] - cost[i+1] + gas[i+2] - cost[i+2] + gas[i+3] - cost[i+3] \geq 0$

To prove → $gas[i+1] - cost[i+1] + gas[i+2] - cost[i+2] + gas[i+3] - cost[i+3] \geq 0$

$prefix[i+3] >= prefix[i]$

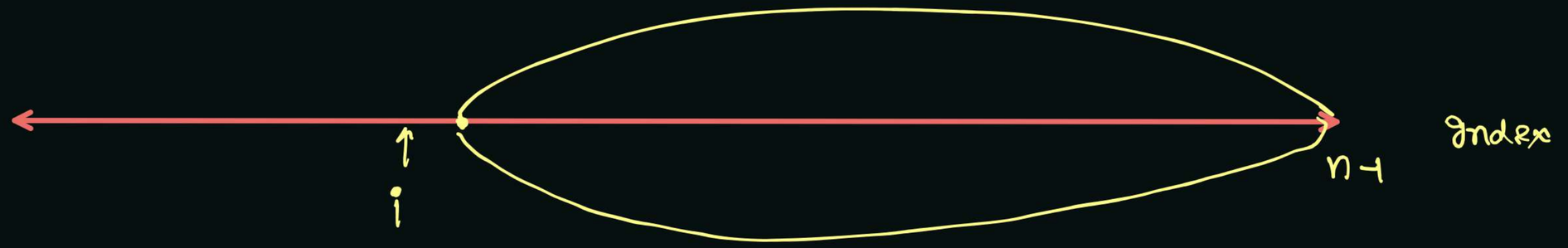
$= prefix[i+2] + gas[i+3] - cost[i+3] \geq prefix[i]$

$= prefix[i+1] + gas[i+2] - cost[i+2] + gas[i+3] - cost[i+3] \geq prefix[i]$

$= prefix[i] + gas[i+1] - cost[i+1] + gas[i+2] - cost[i+2] + gas[i+3] - cost[i+3] \geq prefix[i]$

$\Rightarrow gas[i+1] - cost[i+1] + gas[i+2] - cost[i+2] + gas[i+3] - cost[i+3] \geq 0$

proved



$prefix[n-1] \geq prefix[i]$

$\Rightarrow$ $gas[i+1] - cost[i+1] + gas[i+2] - cost[i+2] + \dots + gas[n-1] - cost[n-1] \geq 0$

NOTE → ① So it is proved that from $(i+1)$ we can move from $i+1$ to n with buffer fuel.

② Now we are at 0^{th} index

can we Reach from 0^{th} Index to 'i' ??

Important → $\sum gas - \sum cost \geq 0$

Already proved → $gas[i+1] - cost[i+1] + gas[i+2] - cost[i+2] + \dots + gas[n-1] - cost[n-1] \geq 0$

we know that $\sum gas - \sum cost \geq 0$

$gas[0] - cost[i] + gas[1] - cost[1]$

$+ gas[2] - cost[2] + \dots + gas[i] - cost[i]$

Part I

$+$

Buffer

$gas[i+1] - cost[i+1] + gas[i+2]$

$- cost[i+2] + \dots + gas[n-1] - cost[n-1]$

Part-II

if its ≥ 0 then we can complete cycle path.

$= [gas[0] + gas[1] + gas[2] + \dots gas[n-1]] - [cost[0] + cost[1] + cost[2] + \dots cost[n-1]]$

$[\sum gas - \sum cost \geq 0]$

part I + part II ≥ 0 → buffer

Sum of SubSeq. Width

Saturday, 18 September 2021 10:31 AM

width → difference b/w min and max.

arr → {2 1 3}

all possible subseq.

Subseq.

				width
0 0 0	→	- - -] →	-	= 0
0 0 1		2	2 - 2	= 0
0 1 0		1	1 - 1	= 0
0 1 1		1 2	2 - 1	= 1
1 0 0		3	3 - 3	= 0
1 0 1		2 3	3 - 2	= 1
1 1 0		1 3	3 - 1	= 2
1 1 1		2 1 3	3 - 1	= 2

Sum = Result

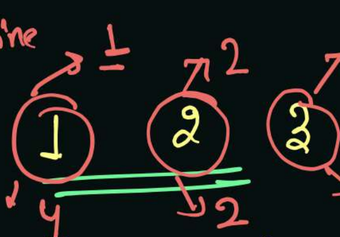
Sum = 6

1*1 + 2*2 + 3*4 - 4*1

- 2*2 - 1*3

= 1+4+12 - 4-4-3

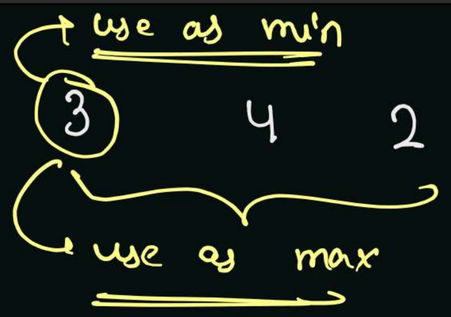
= 17 - 11 = 6



1	→	1 - 1	= 0
2	→	2 - 2	= 0
1 2	→	2 - 1	= 1
3	→	3 - 3	= 0
1 3	→	3 - 1	= 2
2 3	→	3 - 2	= 1
1 2 3	→	3 - 1	= 2

Sum = 6

NOTE: In Subseq. we can skip middle elem



$\equiv$

For width finding -

[2 3 4] → sorted order

max - min = width.

We are finding subseq. so no diff

Subseq.

0 0 0	→	-	=	-
0 0 1	→	- - 2	=	2
0 1 0	→	- 4 -	=	4
0 1 1	→	- 4 2	=	4, 2
1 0 0	→	3 - -	=	3
1 0 1	→	3 - 2	=	3, 2
1 1 0	→	3 4 -	=	3, 4
1 1 1	→	3 4 2	=	3, 4, 2

=	[2-2]	= 0
=	[4-4]	= 0
=	[4-2]	= 2
=	[3-3]	= 0
=	[3-2]	= 1
=	[4-3]	= 1
=	[4-2]	= 2

Sum = 6

$3+3+3+3 = 4*3$

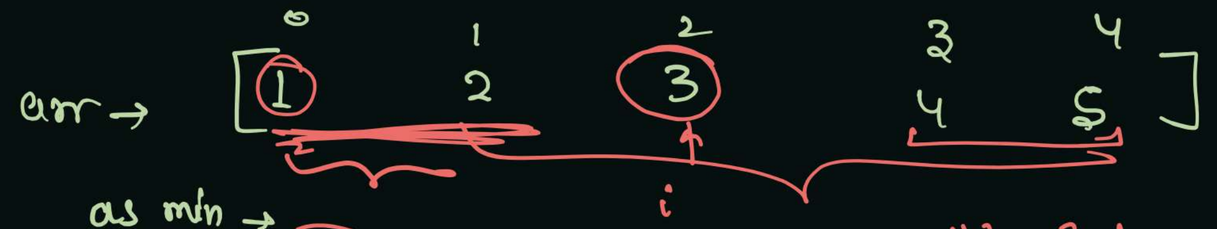
use as min → [2 1 4]

[3, 4, 2]

use as max → [2 4 1]

= Sum of width = $3+3 + 4+4+4+4 + 2 - 2*3$

Sum = $6 + 16 + 2 - 6 - 4 - 8 = 1*4 - 4*2 = 6$



4 - 2

as max → All possible + 'i'

For generic point →

for i^{th} index value.

use as max → 2^{i-0} as max

use as min → 2^{n-1-i}

sum += 2^{i} * 2 - 2^{n-i-1} * 2

arr[i]

2^{i} → max time

2^{n-i-1} → min time

2 → arr[i]

fo $i \rightarrow sum += arr[i] * 2^{i} - arr[i] * 2^{n-i-1}$

arr → [1 2 3 4 5] (indices 0 1 2 3 4)

power → [2^0 2^1 2^2 2^3 2^4] = [1 2 4 8 16]

n=5

n-i	gndx		
4	0	1 * (1 as max) − 1 * (1 as min)	= 1 * 2^0 − 1 * 2^4 = 1 − 16
		+	
3	1	2 * (2 as max) − 2 * (2 as min)	= 2 * 2^1 − 2 * 2^3 = 4 − 16
		+	
2	2	3 * (3 as max) − 3 * (3 as min)	= 3 * 2^2 − 3 * 2^2 = 12 − 12
		+	
1	3	4 * (4 as max) − 4 * (4 as min)	= 4 * 2^3 − 4 * 2^1 = 32 − 8
		+	
0	4	5 * (5 as max) − 5 * (5 as min)	= 5 * 2^4 − 5 * 2^0 = 80 − 5

72 Ans