→ Introduction

→ Working at Razorpay → Razorpayx Engagement Team (Where I am from)
→ Previously worked at Samsung Research → Contacts Application.
→ Graduate from Thapar University 2019 batch
→ Total 3+ years of work experience. 2+ years of experience in mentoring and teaching

Today's content

1) Little Pony and Palindromes
2) Student Marks
3) Another sequence problem
4) Cross the Wall

Little Ponny and Palindromes!

Problem Description

Given a string **A** consisting only of lowercase characters.

You can swap any two characters of the string **A** any number of times, you have to check whether it is possible to convert the string **A** to a palindromic string.

Return **1** if it is possible to else return **0**.

e.g. "0 1 2 3 a a b b" → return 1

→ If we swap 1 and 3 → Palindrome

e.g. "a b a" → return 1.

e.g. "a b c a" →

a → 2

b → 1

c → 1

Observation:-

1) All characters can have even freq.

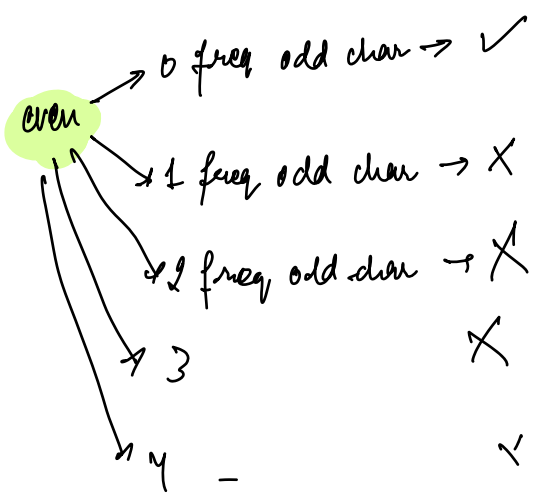
1) MADAM → odd length. → Yes

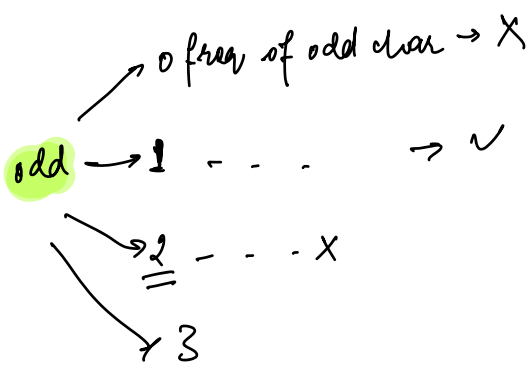
2) MALAYALAM → Yes.

↳ Palindrome

2 1

aba





even
abba

a → 2
b → 2

odd
abcba

a → 2
b → 2
c → 1
d → 3
e → 1

addda ⓒ ⓒ ⓒ ⓧ

d = 3.

Conclusion:-

odd freq characters <= 1

Steps:-

1) Get freq. of all chars in String.
2) int odd = 0.
3) Traverse freq. map and get count of chars with odd freq.
4) if (oddfreq <= 1) return true else return false.

e.g. aabb

a → 2
b → 2

oddfreq = 0 → return true.

e.g. aabbcd → return 0.

a → 2
b → 2
c → 1
d → 1

oddfreq = 2 → return 0.

e.g. aabbccde

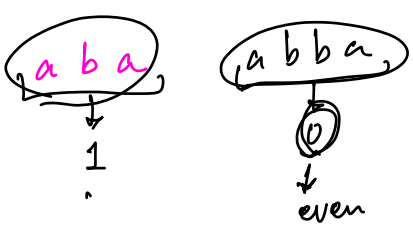
a → 2
b → 2
c → 2
d → 1
e → 1

oddfreq = 2 → return 0.

e.g. abcdecba

a → 2
b → 2
c → 2
d → 1
e → 1

→ return 0



```
if (s.length() % 2 == 0) → even
|     if(oddfreq == 0) return 1;
}
else {
|     if(oddfreq == 1) return 1;
}
return 0;
```

→ if(oddfreq <= 0) return true.

Students Marks

Problem Description

You are given an array of strings **A** representing students name and their score in combined form. For E.g. harsh95 represents harsh has 95 marks.

You have to return another array of strings containing name and their score sorted on the basis of their marks(largest to lowest). If two students have same marks keep the first one from input array in first position.

It is guaranteed that any student's marks do not exceed 100.

Problem Constraints

1 <= A.size() <= 100 → N

1 <= A[i].size() <= 100 → m

e.g. String A[] = { "harsh 95", "jack 100", "john 23", "jess 95" }
(0) (1) (2) (3)

ans [] = { "jack 100", "harsh 95", "jess 95", "john 23" }

↳ size N

idea: find the maximum marks student and place at start.

boolean vis[N] = | T | T | F | F |
(0) (1) (2) (3)

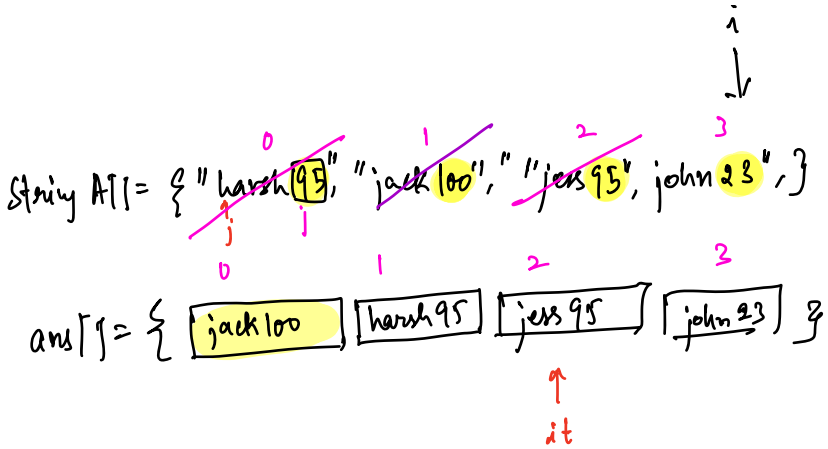
if(vis[i] == true) { continue; }

Pseudo code:-

T.C = O(n*n*m) ; S.C = O(N)

m = string length

```
String ans[N];
bool vis[N];
for(it=0; it<N; it++){
    int max=0, id=-1; //value of max and index of max initialised.
    for(i=0; i<N; i++){
        s=A[i];
        if (vis[i]==true) continue;
        int j=0;
        for(;j<s.length(); j++){
            if (A[i].charAt(j)>='0' && A[i].charAt(j)<='9') break;
        }
        String marks= A[i].substring(j);
        temp= Integer(marks);
        if( temp > max){
            max = temp;
            id = i;
        }
    }
    vis[id] = true;
    ans[it]= A[id];
}
return ans;
```



max=0, id=-1

max=23, id=3.

N log N

↳ Merge Sort

Quick Sort

↳ N log N

Another sequence problem

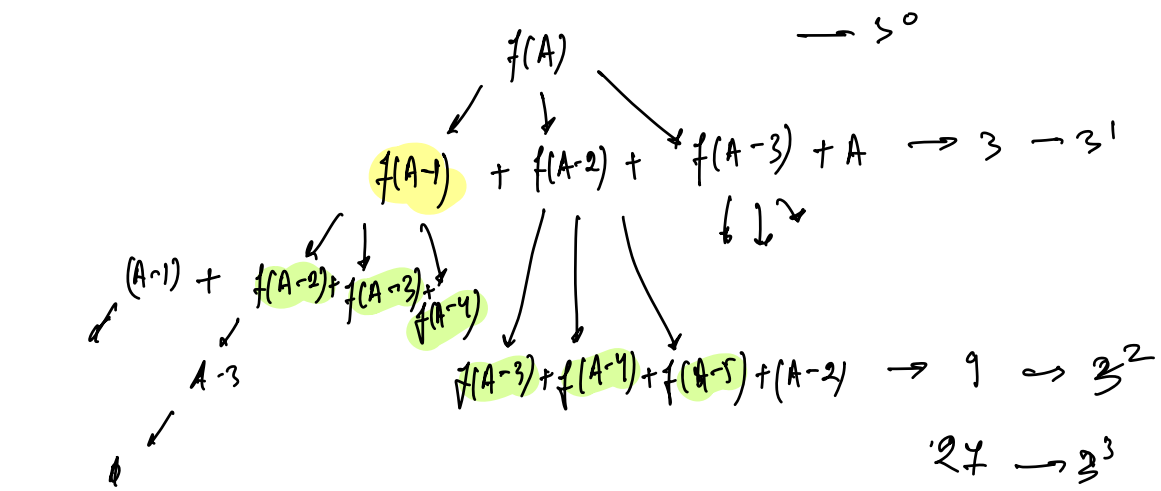
Problem Description

Given a sequence f(A) = f(A-1) + f(A-2) + f(A-3) + A . Calculate the A^{th} term of the sequence.

Given f(0)=1; f(1)=1; f(2)=2;

Problem Constraints

0<=n<=20




```
int main( int A){
    return getValue(A);
}

int getValue( int A){
    if(A==0 || A==1 ) return 1;
    if(A==2) return 2;
    return getValue(A-1) + getValue(A-2) + getValue(A-3) + A
}
```

$0 \le A \le 20$

Assumption :- → Gets a value A and returns f(A)

main logic

f(A-1) f(A-2) f(A-3)

T.C = 3^{An} S.C = O(N)

Cross the wall

Problem Description

There is a rectangular brick wall consisting of several rows of bricks.
The wall has **A** rows, and the length of each row is **B** units. The bricks have the same height that can be considered as 1 unit but has a different length.

You are given an integer array **C** denoting length of **N** bricks.
The bricks are chosen **one by one** from the **left** of the array, and each row of the wall is built from **left to right**.
While building the wall, if the **sum of the length** of bricks in a row is equal to **B**, then start building another row again from **left to right**.

Input is such that you will end up building the wall consisting of **A rows**, and the length of each row will be equal to **B**.

You need to find a vertical line going from top to bottom of the wall that crossed through the fewest number of bricks.
Return the **least number of bricks** through which the vertical line crossed.

NOTE:

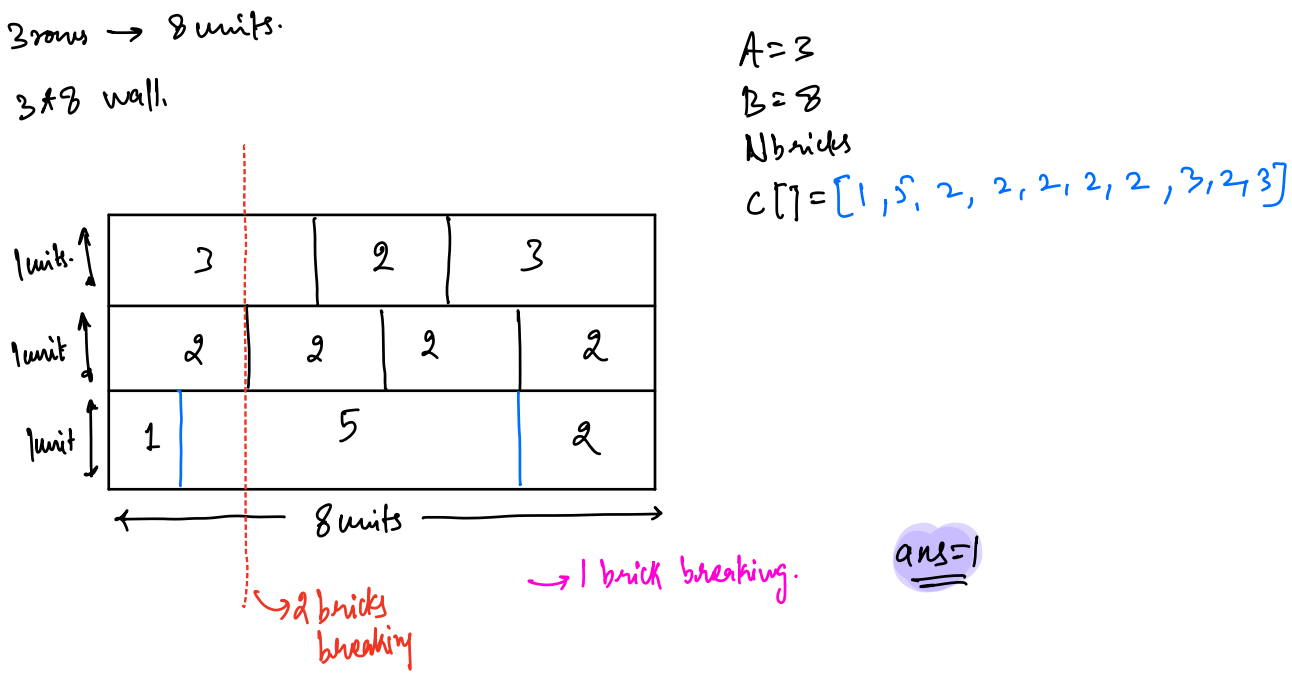
- If your line goes through the edge of a brick, then the brick is not considered as crossed.
- You cannot draw a vertical line just along one of the two vertical edges of the wall, in which case the line will cross no bricks.

Problem Constraints

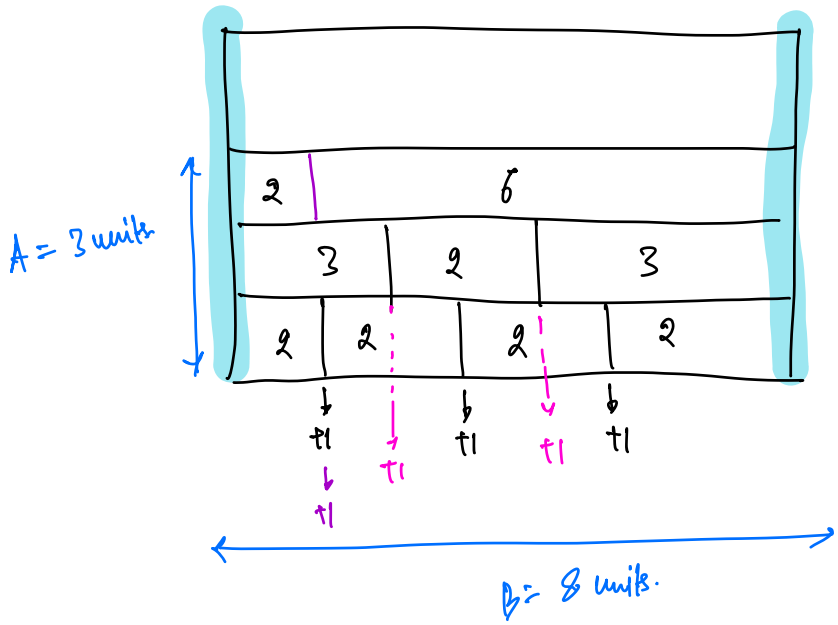
$1 <= N <= 10^5$

$1 <= A \times B <= 10^9$

$1 <= C[i] <= 10^9$



→ We want to find the line that crosses max. no. of edges of the brick.



max = 2

ans = A - max

= 3 - 2 = 1

C = [2, 2, 2, 2, 3, 2, 3, 2, 6]

+1 +1 +1 8%8 == 0

11%8 = 3

13%8 = 5

18%8 = 2

16%8 == 0

24%8 == 0

sum%8 == 0

2 → +1, 2

4 → +1

6 → +1

3 → +1

5 → +1

↳ Map

↳ Use hashmap.

max value.

T.C = O(N)

S.C = O(N)

Final ans = A - maxValue

= 3 - 2

= 1