C. Berry Jam

2 seconds

256 megabytes

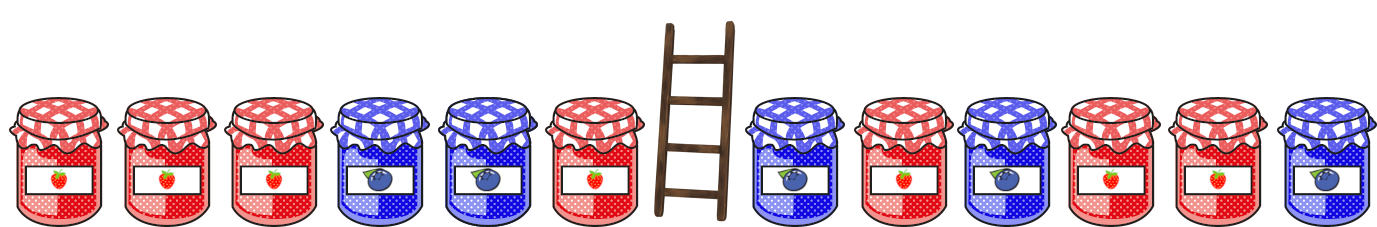
standard input

standard output

Karlsson has recently discovered a huge stock of berry jam jars in the basement of the house. More specifically, there were https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/400/0032.png?V=2.7.5https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Math/Italic/400/006E.png?V=2.7.52n jars of strawberry and blueberry jam.

All the https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/400/0032.png?V=2.7.5https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Math/Italic/400/006E.png?V=2.7.52n jars are arranged in a row. The stairs to the basement are exactly in the middle of that row. So when Karlsson enters the basement, he sees exactly https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Math/Italic/400/006E.png?V=2.7.5n jars to his left and https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Math/Italic/400/006E.png?V=2.7.5n jars to his right.

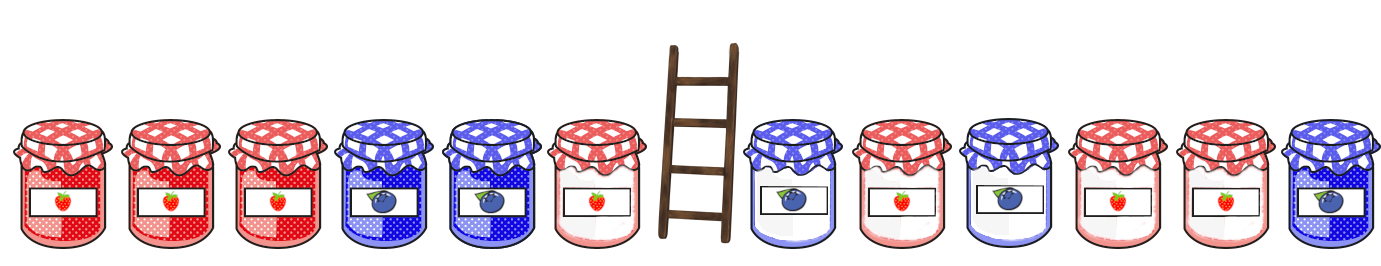
For example, the basement might look like this:



Being the starightforward man he is, he immediately starts eating the jam. In one minute he chooses to empty either the first non-empty jar to his left or the first non-empty jar to his right.

Finally, Karlsson decided that at the end the amount of full strawberry and blueberry jam jars should become the same.

For example, this might be the result:

*He has eaten*https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/400/0031.png?V=2.7.51*jar to his left and then*https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/400/0035.png?V=2.7.55*jars to his right.* *There remained exactly*https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/400/0033.png?V=2.7.53*full jars of both strawberry and blueberry jam.*

Jars are numbered from https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/400/0031.png?V=2.7.51 to https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/400/0032.png?V=2.7.5https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Math/Italic/400/006E.png?V=2.7.52n from left to right, so Karlsson initially stands between jars https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Math/Italic/400/006E.png?V=2.7.5n and https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Math/Italic/400/006E.png?V=2.7.5https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/400/002B.png?V=2.7.5https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/400/0031.png?V=2.7.5n+1.

What is the minimum number of jars Karlsson is required to empty so that an equal number of full strawberry and blueberry jam jars is left?

Your program should answer https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Math/Italic/400/0074.png?V=2.7.5t independent test cases.

**Input**

The first line contains one integer https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Math/Italic/400/0074.png?V=2.7.5t (https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/400/0031.png?V=2.7.5https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/400/2264.png?V=2.7.5https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Math/Italic/400/0074.png?V=2.7.5https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/400/2264.png?V=2.7.5https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/400/0031.png?V=2.7.5https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/400/0030.png?V=2.7.5https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/400/0030.png?V=2.7.5https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/400/0030.png?V=2.7.51≤t≤1000) — the number of test cases.

The first line of each test case contains a single integer https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Math/Italic/400/006E.png?V=2.7.5n (https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/400/0031.png?V=2.7.5https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/400/2264.png?V=2.7.5https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Math/Italic/400/006E.png?V=2.7.5https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/400/2264.png?V=2.7.5https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/400/0031.png?V=2.7.5https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/400/0030.png?V=2.7.5https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/283/0035.png?V=2.7.51≤n≤105).

The second line of each test case contains https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/400/0032.png?V=2.7.5https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Math/Italic/400/006E.png?V=2.7.52n integers https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Math/Italic/400/0061.png?V=2.7.5https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/283/0031.png?V=2.7.5https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/400/002C.png?V=2.7.5https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Math/Italic/400/0061.png?V=2.7.5https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/283/0032.png?V=2.7.5https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/400/002C.png?V=2.7.5https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/400/2026.png?V=2.7.5https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/400/002C.png?V=2.7.5https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Math/Italic/400/0061.png?V=2.7.5https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/283/0032.png?V=2.7.5https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Math/Italic/283/006E.png?V=2.7.5a1,a2,…,a2n (https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/400/0031.png?V=2.7.5https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/400/2264.png?V=2.7.5https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Math/Italic/400/0061.png?V=2.7.5https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Math/Italic/283/0069.png?V=2.7.5https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/400/2264.png?V=2.7.5https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/400/0032.png?V=2.7.51≤ai≤2) — https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Math/Italic/400/0061.png?V=2.7.5https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Math/Italic/283/0069.png?V=2.7.5https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/400/003D.png?V=2.7.5https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/400/0031.png?V=2.7.5ai=1 means that the https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Math/Italic/400/0069.png?V=2.7.5i-th jar from the left is a strawberry jam jar and https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Math/Italic/400/0061.png?V=2.7.5https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Math/Italic/283/0069.png?V=2.7.5https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/400/003D.png?V=2.7.5https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/400/0032.png?V=2.7.5ai=2 means that it is a blueberry jam jar.

It is guaranteed that the sum of https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Math/Italic/400/006E.png?V=2.7.5n over all test cases does not exceed https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/400/0031.png?V=2.7.5https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/400/0030.png?V=2.7.5https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/283/0035.png?V=2.7.5105.

**Output**

For each test case print the answer to it — the minimum number of jars Karlsson is required to empty so that an equal number of full strawberry and blueberry jam jars is left.

**Example**

**input**

**Copy**

4

6

1 1 1 2 2 1 2 1 2 1 1 2

2

1 2 1 2

3

1 1 1 1 1 1

2

2 1 1 1

**output**

**Copy**

6

0

6

2

**Note**

The picture from the statement describes the first test case.

In the second test case the number of strawberry and blueberry jam jars is already equal.

In the third test case Karlsson is required to eat all https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/400/0036.png?V=2.7.56 jars so that there remain https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/400/0030.png?V=2.7.50 jars of both jams.

In the fourth test case Karlsson can empty either the second and the third jars or the third and the fourth one. The both scenarios will leave https://assets.codeforces.com/mathjax/fonts/HTML-CSS/TeX/png/Main/Regular/400/0031.png?V=2.7.51 jar of both jams.

#include<bits/stdc++.h>

#define pb push\_back

#define pii pair<int,int>

#define int long long int

#define vec vector<int>

#define inf 1e18

using namespace std;

int32\_t main()

{

ios\_base::sync\_with\_stdio(false);

cin.tie(NULL);

cout.tie(NULL);

int tt=1;

cin>>tt;

while(tt--)

{

int n;

cin>>n;

n\*=2;

int a[n+1],i;

for(i=1;i<=n;i++)

cin>>a[i];

int o=0,z=0;

int pr[n+1]={0};

map<int,int> h,h2;

for(i=1;i<=n;i++)

if(a[i]==1)

o++;

else

z++,a[i]=-1;

for(i=1;i<=n;i++)

pr[i]+=pr[i-1]+a[i];

for(i=n/2;i<=n;i++)

if(h[pr[i]]==0)

h[pr[i]]=i;

for(i=n/2;i>=0;i--)

if(h2[pr[i]]==0)

h2[pr[i]]=i;

int ans=inf;

o-=z;

for(i=0;i<=n/2;i++)

{

if(h[pr[i]+o]>i)

ans=min(ans,h[pr[i]+o]-i);

}

if(!o)

ans=min(ans,o);

cout<<ans<<"\n";

}

}

Learnings

1. We can find subarray sum with a given condition that we need to take a particular index in the subarray.
2. We can use prefix sum technique, fixing left index and binary searching the right index.
3. Time complexity is O(nlogn).