

Solution

A

将所给数组分成 m 个区间，使得每个区间满足

$$2^0 \cdot a_i < 2^1 \cdot a_{i+1} < \dots < 2^x \cdot a_{i+x}$$

如果原数组存在 $a[i-1] \geq a[i] \cdot 2$ 表明 $a[i]$ 属于下一个这样的区间，给每个区间标号 $0-m-1$

枚举 $s[i] == s[i+k]$ 表明存在一个答案

Code

```
#include <iostream>
using namespace std;
int main()
{
    int t;
    cin >> t;
    while (t--)
    {
        int n, k;
        scanf("%d %d", &n, &k);
        int a[n+1];
        a[0] = 0;
        int s[n+1];
        s[0] = s[1] = 0;
        int ans = 0;

        for(int i=1;i<=n;i++) scanf("%d", &a[i]);
        for(int i=1;i<=n;i++)
        {
            s[i] = s[i-1];
            if(a[i-1] >= a[i] * 2) s[i] ++;
        }
        for(int i=1;i<=n-k;i++) if(s[i] == s[i+k]) ans ++;
        cout << ans << endl;
    }

    return 0;
}
```

B

首先把 $a_i \geq i$ 的舍弃，然后我们构造一个 b 数组 $b[i]$ 表示下标大于 i 的元素中，值大于 i 的个数（也就是 a_i 后面有多少个 a_j 满足 $a_i < i < a_j < j$ ）

然后枚举所有 $a[i]$ ，对于任意一个 $a[i]$ 都会使b数组区间 $[1, a[i] - 1]$ 每个数减一，那么先把b数组当做差分数组，对其进行操作，最后再复原成前缀和数组，b数组所有元素的和就是答案

Code

```
#include <iostream>
#include <cstring>
#include <algorithm>
#include <vector>
#include <queue>
#define inf 0x3f3f3f3f
using namespace std;
typedef long long ll;
typedef unsigned long long ull;
typedef pair<int, int> pii;
const int N = 2e5+10;
int a[N], b[N];
int main()
{
    int t;
    cin >> t;
    while (t--)
    {
        memset(b, 0, sizeof(b));
        int n;
        scanf("%d", &n);
        for(int i=1; i<=n; i++)
        {
            scanf("%d", &a[i]);
            if(a[i] >= i) a[i] = -1;
        }
        for(int i=1; i<=n; i++)
        {
            if(a[i] == -1) continue;
            else b[a[i]] --, b[1] ++;
        }
        for(int i=1; i<=n; i++) b[i] += b[i-1];
        ll ans = 0;
        for(int i=1; i<=n; i++)
        {
            if(a[i] == -1) continue;
            ans += b[i];
        }
        printf("%lld<br>", ans);
    }

    return 0;
}
```

构造一个mod 7的前缀和数组, 如果 $s[i] == s[j]$ 那么区间 $[i + 1, j]$ 的和必然是7的倍数
然后枚举0-6, 找到其中最长的即可

Code

```
#include <iostream>
using namespace std;
const int N = 50010;
int a[N], s[N];
int main()
{
    int n;
    cin >> n;
    for(int i=1; i<=n; i++) scanf("%d", &a[i]);
    for(int i=1; i<=n; i++) s[i] = (s[i-1]+a[i]) % 7;
    int ans = 0;
    for (int i = 0; i < 7; i++)
    {
        int l = 0, r = n;
        while(s[l] != i && l <= n) l ++;
        while(s[r] != i && r >= 1) r --;
        ans = max(ans, r - l);
    }

    cout << ans;
    return 0;
}
```

D

一维差分模板题

Code

```
#include <iostream>
using namespace std;
const int N = 5e6+10;
int a[N], s[N];
int main()
{
    int n, p;
    cin >> n >> p;
    for(int i=1; i<=n; i++)
    {
        int x;
        scanf("%d", &x);
        a[i] += x, a[i+1] -= x;
    }
    while (p--)
    {
```

```

        int x, y, z;
        scanf("%d %d %d", &x, &y, &z);
        a[x] += z, a[y+1] -= z;
    }
    for(int i=1;i<=n;i++) s[i] = s[i-1] + a[i];
    int ans = 1e9;
    for(int i=1;i<=n;i++) ans = min(s[i], ans);
    cout << ans << endl;
    return 0;
}

```

E

题目可以转化为，使得原数组的差分数组前半部分大于0，后半部分小于0

分别定义两个指针 l, r ，其中， l 指向从头数第一个不大于0的数， r 指向从后往前数第一个不小于0的数
依次更改循环，直到 $l \geq r$

Code

```

#include <iostream>
using namespace std;
typedef long long ll;
const int N = 2e5+10;
ll a[N], s[N];
int n;
int main()
{
    cin >> n;
    for(int i=1;i<=n;i++)
    {
        int x;
        scanf("%d", &x);
        a[i] += x, a[i+1] -= x;
    }
    int l = 1, r = n;
    ll ans = 0;
    while(l<r)
    {
        while(a[l]>0) l++;
        while(a[r]<0) r--;

        if(l>r) break;
        int d = min(1-a[l], a[r]+1);
        ans += d;
        a[l] += d;
        a[r] -= d;
    }
    cout << ans;
    return 0;
}

```

