

题解

题目分析

这题真的water

我们可以从二进制角度，一位一位地考虑。

可以发现若有奇数个连续和的第 k 位为1，那么答案的第 k 位为1，否则为0。

因此可以想到用前缀和，当 $s[i]$ 与 $s[j]$ ($0 < j < i$) 的差的第 k 位为1时，这段连续和对答案的第 k 位有贡献。考虑第 i 个前缀和 $s[i]$ 的第 k 位：

若这一位为1，那么 $s[j]$ 需要满足第 k 位为0 且前 $k-1$ 位比 $s[i]$ 前 $k-1$ 位小 或者 第 k 位为1 且前 $k-1$ 位比 $s[i]$ 前 $k-1$ 位大（错位相减）。

若这一位为0，同理。

所以我们可以用两个树状数组分别维护 $1 \dots \max n$ 的数中第 k 位为1和0的个数，然后就可以求出答案的每一位了。

代码

```
#include<iostream>
#include<cstdio>
#include<cstring>
#define lowbit(x) (x&(-x))
using namespace std;

int n,ans=0,a[100010],b[1000100][2],rest[100010];

void insert(int pos,int o){
    for(;pos<=1000010;pos+=lowbit(pos))b[pos][o]++;
}

int query(int pos,int o){
    int ret=0;
    for(;pos>0;pos-=lowbit(pos))ret+=b[pos][o];
    return ret;
}
```

```

}

int main()
{
    scanf("%d",&n);
    for(int i=1;i<=n;i++)scanf("%d",&a[i]),a[i]+=a[i-1];
    for(int i=0;i<=20;i++){
        if((1<<i)>a[n])break;
        memset(b,0,sizeof(b));
        int t,count=0;
        insert(1,0);
        for(int j=1;j<=n;j++){
            bool o=a[j]&(1<<i);
            t+=query(rest[j]+1,o^1);
            t+=query(1000010,o)-query(rest[j]+1,o);
            insert(rest[j]+1,o);
            if(o)rest[j]|=(1<<i);
            if(t&1)count^=1;
        }
        if(count)ans|=(1<<i);
    }
    printf("%d",ans);
    return 0;
}

```