

# 题解

## 0x00 理解题目

这个题说人话就是，对于每一个子区间计算一个权值：这个子区间每个元素的和乘上这个区间元素最大值。最后

## 0x01 解题

### 00 猜想

这个题目直接计算显然爆炸，过不了。我们观察这个权值可以发现，其中一个因子是**区间和**，这个东西我们可以前缀和， $O(1)$  求出；而另外一个因子是区间最大值，这个东西我们可以用区间最值算法  $O(1)$  查询求出，现在时间复杂度已经做到了  $O(n^2)$ 。但是我们需要进一步优化，可以见到的是前缀和这一堆东西可以进一步抽象成一些东西，并且还能  $O(1)$  求出。那么剩下的这一部分就是最大值的部分如何处理，现在剩下的问题就是区间最大值的因子。很明显的是，有子区间跨过了这个区间的最大值，那么这些子区间的最大值因子都是一样的，如先前所说，我们猜想：这些区间的区间和简化一下式子或许可以  $O(1)$  全部一次性求出；现在观察剩下的这些区间，注意到所有的这些子区间被分成了两部分，区间最值左边部分和区间最值右边部分，我们单独将这两部分的权值和单独处理出来就可以了，惊奇的发现，求解这两部分小区间的权值和和求解大区间的权值和本质是一个问题，那么就可以分治求解了。

### 01 细化

有了先前的思路，我们简化式子就是顺水推舟的一个过程了。为了方便令： $W_{[L,R]}$  表示  $[L, R]$  这个区间的权值和。既：

$$W_{[L,R]} = \sum_{l=L}^R \sum_{r=l}^R \left[ \left( \max_{k=l}^r A_k \right) \times \sum_{j=l}^r A_j \right] \bmod (2^{64} - 1)$$

我们暂且不讨论模数的问题。令  $X$  为这个区间最值下标，按照上一部分所说我们可以将式子化简为：

$$W_{[L,R]} = W_{[L,X]} + W_{[X,R]} + V_{[L,R]} \times A_X$$

其中  $V_{[L,R]}$  表示跨过这个区间最大值的区间权值和，可以观察到，每次递归会将元素个数减少 1。易知，最坏情况只需要递归出来  $2n - 1$  个节点，也就是  $O(n)$  的时间复杂度。剩下部分时间复杂度取决于  $V$  函数的时间复杂度

现在我们只对  $V$  这个函数进行讨论，需要明确的一点是这个函数时间复杂度我们期望做到  $O(1)$ 。那么我们开始推式子：

显然，

$$V_{[L,R]} = \sum_{i=X}^R \sum_{j=L}^X \sum_{k=i}^j A_k$$

对于  $\sum_{k=i}^j A_k$  使用前缀和即可  $O(1)$  求出。令：

$$sum_i = \sum_{j=1}^i A_j$$

$$V_{[L,R]} = \sum_{i=X}^R \sum_{j=L}^X (sum_i - sum_{j-1})$$

$$V_{[L,R]} = \sum_{i=X}^R \sum_{j=L}^X (sum_i) - \sum_{i=X}^R \sum_{j=L}^X (sum_{j-1})$$

$$V_{[L,R]} = \sum_{j=L}^X \sum_{i=X}^R (sum_i) - \sum_{i=X}^R \sum_{j=L}^X (sum_{j-1})$$

$$V_{[L,R]} = (X - L + 1) \sum_{i=X}^R (sum_i) - (R - X + 1) \sum_{j=L}^X (sum_{j-1})$$

到了这一步由于  $\sum_{i=X}^R (sum_i)$  和  $\sum_{j=L}^X (sum_{j-1})$ ，这个公式计算时间复杂度还是  $O(n)$ ，考虑忽略  $sum$  这个名字，实际上这两部分还是在做区间和的问题我们再进行一次前缀和就能  $O(1)$  求出。令：

$$sumsum_i = \sum_{j=1}^i sum_j$$

显然

$$V_{[L,R]} = (X - L + 1) (sumsum_R - sumsum_{X-1}) - (R - X + 1) \sum_{j=L}^X (sumsum_{X-1})$$

那么求出答案时间复杂度为  $O(n + m)$ ，其中  $O(m)$  为处理区间最值得时间复杂度。可以做到  $O(n \log n)$  预处理  $O(1)$  查询。

## 0x02 代码实现

---

不能直接递归会爆栈。

```

#include<stdio>
#define file
#define INPUT_DATA_TYPE int
#define OUTPUT_DATA_TYPE unsigned long long
INPUT_DATA_TYPE read(){register INPUT_DATA_TYPE x=0;register char
f=0,c=getchar();while(c<'0' || '9'<c)f=(c=='-'),c=getchar();while('0'<=c&&c<='9')x=
(x<<3)+(x<<1)+(c&15),c=getchar();return f?-x:x;}void print(OUTPUT_DATA_TYPE x)
{register char s[20];register int i=0;if(x<0){x=-x;putchar('-');}if(x==0)
{putchar('0');return;}while(x){s[i++]=x%10;x/=10;}while(i){putchar(s[--
i]+'0');}return;}

int arr[500010];
unsigned long long sum[500010],sumsum[500010],ans;

struct stresult{
    int data,index;
};

#include<cstring>
#define M >
#define mm(a,b) ((a)<M(b)?(a):(b))
#define ST_DATA_TYPE int
int ln(int n){
    return 63-__builtin_clzll(n);
}
const int ST_MAX_I=500010;
const int ST_MAX_J=20;
struct ST{
    ST_DATA_TYPE f[ST_MAX_I][ST_MAX_J]; //f[i][j]---[i,i+2^j-1]
    ST_DATA_TYPE book[ST_MAX_I][ST_MAX_J];

    void build(ST_DATA_TYPE *data,int size){
        memset(f,0,sizeof(f));
        int k=ln(size);
        for(register int i=1;i<=size;++i){
            f[i][0]=data[i];
            book[i][0]=i;
        }

        for(register int j=1;j<=k;++j){
            for(register int i=1;i<=size-(1<<j)+1;++i){
                if(f[i][j-1]>f[i+(1<<(j-1))][j-1]){
                    f[i][j]=f[i][j-1];
                    book[i][j]=book[i][j-1];
                }else{
                    f[i][j]=f[i+(1<<(j-1))][j-1];
                    book[i][j]=book[i+(1<<(j-1))][j-1];
                }
            }
        }
    }
};

```

```

        return;
    }

    stresult query(int l,int r){
        int k=ln(r-l+1);
        if(f[l][k]>f[r-(1<<k)+1][k]){
            return (stresult){f[l][k],book[l][k]};
        }
        return (stresult){f[r-(1<<k)+1][k],book[r-(1<<k)+1][k]};
    }
}st;

struct stack_node{
    int l,r;
    char flag;
    stresult str;
    stack_node(int inl,int inr){
        l=inl;
        r=inr;
        flag=0;
        str=st.query(l,r);
        ans+=V();
        return;
    }

    unsigned long long V(){
        return ((str.index-l+1)*(sumsum[r+1]-sumsum[str.index])-(r-str.index+1)*
(sumsum[str.index]-sumsum[l-1]))*str.data;
    }

    stack_node(){
        return;
    }
};

#define STACK_DATA_TYPE stack_node
const int STACK_SIZE=500010;struct STACK{STACK_DATA_TYPE _null,s[STACK_SIZE];int
s_top;STACK(){s_top=0;return;}void push(STACK_DATA_TYPE in)
{s[s_top]=in;++s_top;return;}void pop(){--s_top;return;}void pop(int in){s_top-
=in;return;}STACK_DATA_TYPE top(){return s[s_top-1];}void clear(){s_top=0;}bool
empty(){if(s_top==0){return true;}else{return false;}}int size(){return
s_top;}STACK_DATA_TYPE read(int in){return s[in];}};
STACK S;

int main(){
    #ifdef file
    freopen("sequence.in", "r", stdin);
    freopen("sequence.out", "w", stdout);
    #endif

```

```

register int i;
int n=read();
for(i=1;i<=n;++i) arr[i]=read();
for(i=1;i<=n;++i) sum[i]=sum[i-1]+arr[i];
for(i=1;i<=n;++i) sumsum[i+1]=sumsum[i]+sum[i];
st.build(arr,n);

register stack_node now(1,n),temp;
S.push(now);
while(!S.empty()){
    now=S.top();
    S.pop();
    if(now.flag==2||now.l>now.r) continue;
    else if(!now.flag)
        temp=stack_node(now.l,now.str.index-1);
    else if(now.flag==1)
        temp=stack_node(now.str.index+1,now.r);
    ++now.flag;
    S.push(now);
    S.push(temp);
}

print(ans);

#ifdef file
fclose(stdin);
fclose(stdout);
#endif
return 0;
}

```