

线段树基础选讲

东营市第一中学 孙翊轩

分治算法初探

解决以下问题：

N 次询问，每次询问给出 A, B 与 P 。
对于每次询问，求出 $A^B \bmod P$ 。

其中 $N \in [1, 100] \cap \mathbb{Z}, A, B \leq P \in [1, 10^9] \cap \mathbb{Z}$ 。

分治算法初探

- 考虑一个朴素算法：

$A^B = A * A * \dots * A$ ，只要每次模 P 即可，时间复杂度为 $O(n)$ 。

```
int MyPow(int A,int B,int P){  
    int base=A;  
    for(int i=2;i<=B;++i)  
        base=(base*A)%P;  
    return base;  
}
```

分治算法初探

- 分治
顾名思义为分而治之。

再考虑刚才的问题，可以发现如下的性质：
对于求 A^B ，只需求 $A^{B/2}$ 然后再相乘即可。
对于求 $A^{B/2}$ ，只需求 $A^{B/4}$ 然后再相乘即可。
那么接下来问题的都可以进一步划分了。

这样做的复杂度是 $O(\log_2 N)$ 的，大家学到对数就可以证明了。

分治算法初探

- 刚才的算法可以容易的使用递归实现
 - 注意观察代码中对于指数为奇数的位置的处理方式

```
int FastPow(int A,int B,int P){  
    if(B==1)return A;  
    int F=FastPow(A,B>>1,P);  
    return (B&1?(F*F*A):F*F)%P;  
}
```

分治算法初探

- 总结

大家可以发现分治思想是把大事化小，小事化了，最终合并答案求得结果。

其宗旨是：我要的让我的下级来求，具体怎么求我不管。

在讲线段树之前说到分治算法的目的是为了让大家理解分治的思想，能明白递归具有轻松实现分治的优美性质。

线段树的优美性质实际上就是基于分治的优美性质而来。

线段树引入

- 引入
 - 今天我们不从问题引入线段树，而是选择从数据结构的角度引入线段树。
 - 和数组一样，线段树也是一种数据结构。

顾名思义，数据结构就是存储数据的结构。

而线段树和数组，都是解决线性数据问题的数据结构。

线性数据结构的常见操作

常见操作	单点修改	单点查询	区间查询	区间加减	区间乘除	区间取模	区间开方
数组	$O(1)$	$O(1)$	$O(N)$	$O(N)$	$O(N)$	$O(N)$	$O(N)$
线段树	$O(\log_2 N)$	$O(\log_2 N)$	$O(\log_2 N)$	$O(\log_2 N)$	$O(\log_2 N)$	$O(2 \log_2 N)$	$O(\log_2 N)$

- 明晰
 - 上述常见操作用数组如何实现以及复杂度如何推导
- 思考
 - 线段树比数组的优势在何？
 - 线段树一定比数组快吗？
 - 什么情况下需要使用线段树？

线段树

- 线段树

大家未见其人先闻其名，刚才已经介绍过了线段树的优秀复杂度。但是线段树博大精深，刚才的有些操作实际上是很复杂的。

今天我们讲的线段树将会解决以下问题：

在一个线性数据结构中，进行如下操作：

- 单点修改（这里是修改一个数字）
- 单点查询（这里是查询一个数字的值）
- 区间加减（这里是对区间内所有数字加减一个数）
- 区间查询（这里是查询一个区间内数字的总和）