数据结构

ST 表

ST表是利用倍增思想做预处理的一种数据结构，而**预处理**所指的算法自然就是**动态规划**，表示状态和转移状态。

设 $f(i,j)$ 代表第 i 个数到第 $i+2^j-1$ 个数的最大值。

即 $f(i,j)=\max(i,i+2^j-1)$

而我们可以发现 $f(i,j)$ 所管辖的区间取决于 $[i,i+2^{j-1}-1],[i+2^{j-1},i+2^j-1]$ 这两个区间。

可得转移公式：$f(i,j)=\max(f(i,j-1),f(i+2^{j-1},j-1))$

```
int f[200000][20];
void build() {
    for(int j = 1; j <= logn; j ++)
        //j + (1 << i) - 1 为右端点
        for(int i = 1; i + (1 << j) - 1 <= n; i ++)
            f[i][j] = max(f[i][j - 1], f[i + (1 << (j - 1))][j - 1]);
}
```

也就是说，我们可以通过这个转移式子在 $n\cdot\log_2 n$ 的时间范围内预处理出所有 $f(i,j)$。

那么我们如何查询呢?

首先我们得知道，查询区间的重叠这个操作是不会影响到最大值的。
对于一个数组来说， 求其区间 $[1,10]$ 的最大值，可由区间 $[1,7]$ 和区间 $[2,10]$ 的最大值得出。

那么我们只需要维护我们查询的区间在 $[l,r]$ 内即可，还记得 $f(i,j)$ 的含义吗?

查询区间 $[l,r]$ 可转为两个 $f(i,j)$ 的最大值，那么我们访问的 $f(i,j)$ 里面的数应该是什么呢?

首先，我们得保证我们查询的值不会超过 $r-l+1$ 这个区间大小，所以可以先求出它的 $q=\lfloor\log_2(r-l+1)\rfloor$，而后我们可以得到第一个式子为 $f(i,q)$

第二个式子得保证它刚好碰到 r ，也就是说 $r = x + 2^j - 1$， 移项可以得到第二个式子 $f(r - 2^j + 1, q)$

将这两个式子合起来，即为区间 $[l, r]$ 的最大值，访问两个数组元素，即为 $O(1)$ 的时间复杂度。

$$q = \lfloor \log_2(r - l + 1) \rfloor$$
$$\max(l, r) = \max(f(i, q), f(r - 2^j + 1, q))$$

如何求出合适的 $\lfloor \log_2 \rfloor$ 呢?

```
int lg[maxn];
void Log() {
    lg[1] = 0, lg[2] = 1;

    for(int i = 3; i <= maxn; i++)
        lg[i] = lg[i >> 1] + 1;
}
// or
#include <cmath>
floor(log2(233))
```