

RMQ、ST、跳表

算法分析:

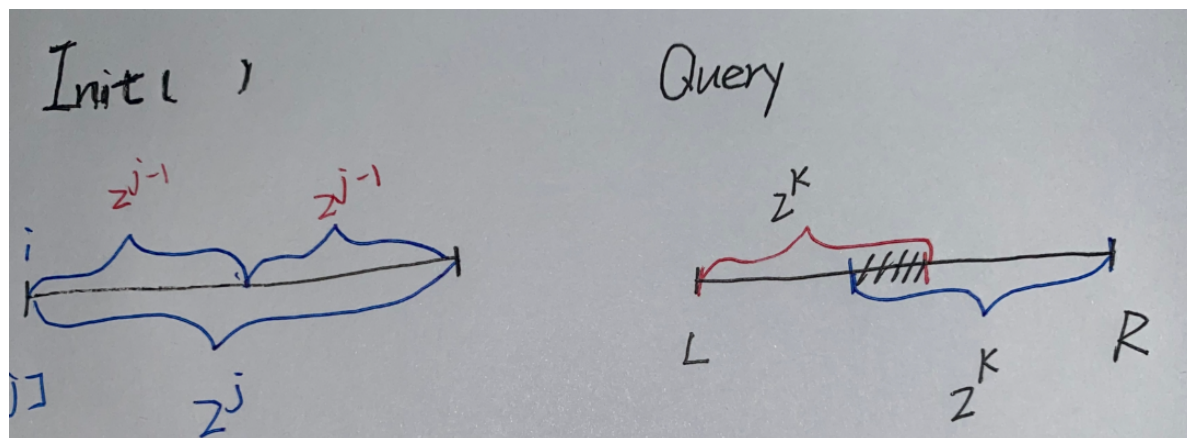
时间复杂度: $n * \log n$

$f[i][j]$ 表示:从 i 为左端点,长度为 2^j 的区间内的 最大值/最小值

思路:

- 初始化:将长度为 2^j 的区间划分成两个 2^{j-1} 的, 得到递推式 $f[i][j] = \max(f[i][j-1], f[i+(1 \ll j-1)][j-1])$
- 查询:求最大的 $2^k \leq$ 区间长度的 k , 取 $[L, 2^k]$ 和 $[R - 2^k + 1, R]$ (以 R 为中点)

一定有重合因为 $2 * 2^k > \text{len}$, 所以能算出最大值



静态的、缺点是不能修改,不过相比线段树来说要短很多

code

```
int f[N][M];

void init() // 将长度为 2^j 的区间分成两个 2^{j-1} 的区间
{
    for(int j = 0; j < M; j++)
        for(int i = 1; i + (1 << j) - 1 <= n; i++)
            if(!j) f[i][j] = w[i];
            else f[i][j] = max(f[i][j-1], f[i+(1<<j-1)][j-1]);
}

int query(int l, int r)
{
    int len = r - l + 1;
    int k = log(len) / log(2);

    return max(f[l][k], f[r-(1<<k)+1][k]);
}
```

