

I题题解报告:

题意: 求指定区间内所有子区间中最小的异或和

首先预处理一下, 得到从第*i*点异或到第*j*个点的值的大小, 存入数组 **XOR[i][j]** 中。

令 **dp[i][j]** 表示区间[i,j]中子区间最小的异或和。

考虑区间dp, 令len = 1, 然后不断增大len, 更新dp[i][j];

转移方程可以这样写:

```
dp[i][j] = max (XOR[i][j], dp[i+1][j]);  
dp[i][j] = max (dp[i][j], dp[i][j-1]);
```

之所以能够这样写, 最根本的原因就是

我们的区间长度是从小到大逐渐增大的。

设想区间[i,j]有一个超级区间[a,b], 这个超级区间的异或值总是最大的。那么当 len = b - a + 1时, 我们得到dp[a][b]就是XOR[a][b];

len增大, dp[a-1][b]和dp[a][b+1]和dp[a][b]相等;

len继续增大.....

因为区间长度是从小到大更新的, 所以我们能保证更新的dp[i,j]的正确性。

如果事先知道区间dp, 这道题可以直接秒杀的...相比用到的算法, 我反而觉得预处理更有意思些。

```
//一个倒金字塔  
for(int i=2;i<=n;++i){  
    for(int j=1;j<=n-i+1;++j){  
        tmp[i][j]=tmp[i-1][j]^tmp[i-1][j+1];  
    }  
}  
for(int i=1;i<=n;++i){  
    for(int j=i;j<=n;++j){  
        XOR[i][j]=tmp[j-i+1][i];  
    }  
}
```

时间复杂度: $O(n^2)$

区间长度从1到n, 每次都要从第一个点跑到最后一个点