[SCOI2016]萌萌哒

算法： 倍增并查集

$Subtask\ 1$：

不难发现，这是一个区间合并的计数类问题。

于是， 考虑直接暴力合并区间，用**一般的并查集**。

最后，统计出联通块数量。时间复杂度 $O(n^2)$。

$Subtask\ 2$：

观察到，对于并查集的区间合并是有 "结合律" 的。

$$Merge(l1,r1,l2,r2)=Merge(Merge(l3,r3,l4,r4),Merge(l5,r5,l6,r6))$$

因为从小区间合并成大区间，时间复杂度是难以降下来的。

所以可以先合并大区间，然后将大区间拆分成小区间逐层拆分合并，这样时间复杂度是 $nlogn$ 的。

重点是考虑如何不漏的拆分大区间成小区间。(**区间重叠是不影响结果的**)

线段树不容易实现，但是倍增就相对容易许多，结合 ST 表的想法来实现。

`f[i][j]` 表示区间 $[i,i+2^j-1]$ 所在的集合，将其拆分为 $[l,2^{j-1}]$和$[r-2^{j-1}+1,2^{j-1}]$。

在 $pushdown$ 时，因为是从大区间拆分成小区间，故 j 递减。

于是对于起点 i 的区间，$[i,i+2^j-1]$分为$[i,i+2^{j-1}-1]$和$[i+2^{j-1},i+2^j-1]$。

最后统计答案即 $9*10^{(集合数-1)}$。

并查集的合并 (参考代码)：

```
inline int find(int x,int l){
    if(x == f[x][l]) return x;
    else return f[x][l] = find(f[x][l],l);
}
inline void toge(int x,int y,int l){
    int r1=find(f[x][l],l);int r2=find(f[y][l],l);
    if(r1!=r2)f[f[r1][l]][l]=r2;
    return;
}
```