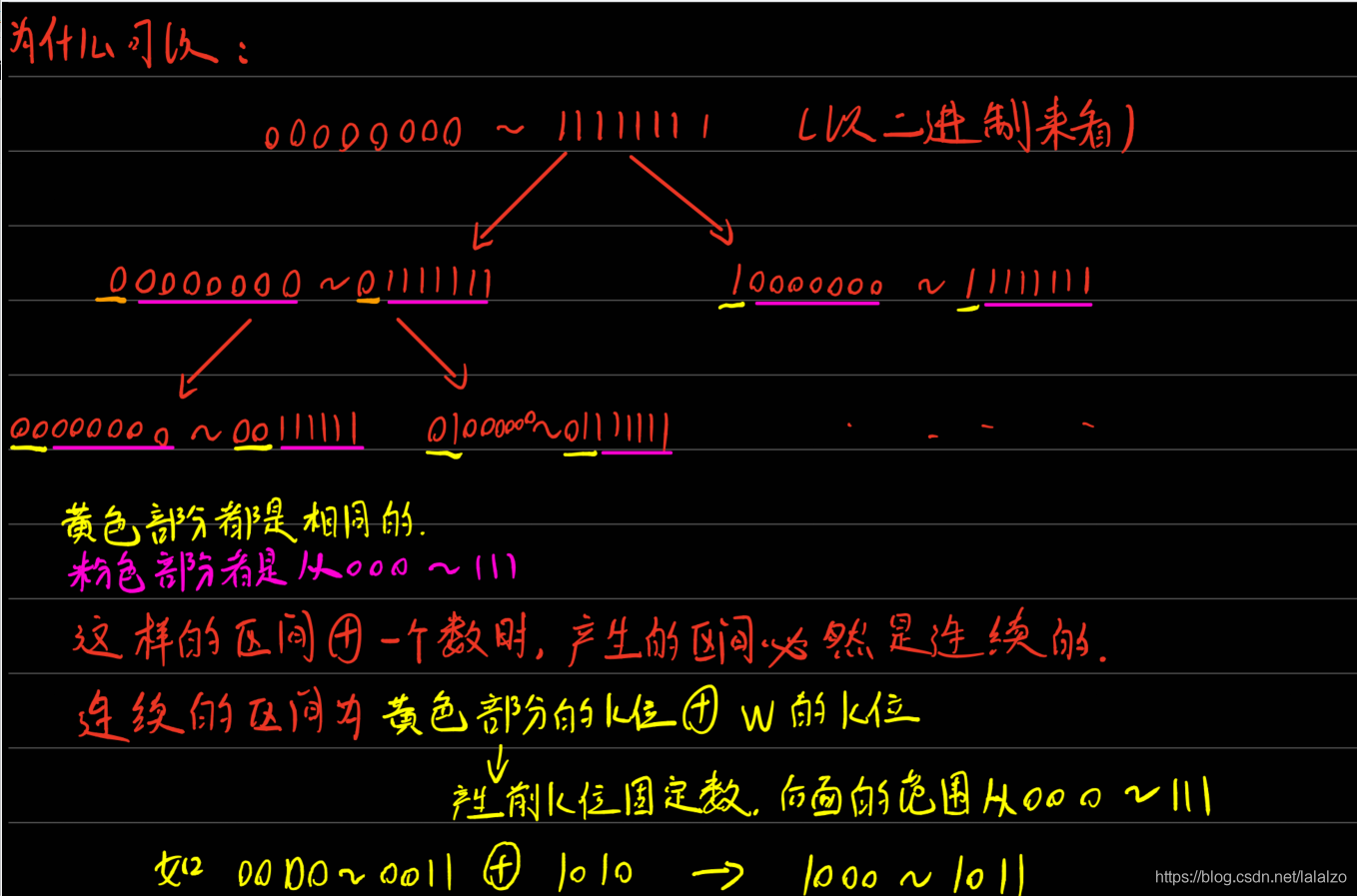
我们将利用线段树 将不连续的区间分为logw段连续的区间。



 做法一：n\*logn 使用线段树可以求出所有不合法区间的并集，把不合法的区间标记为1，可以没有标记过的区间说明对于这n个点这些区间都是合法的。

int ch[N][2],sum[N];bool tag[N];vector<pii> g[N];int l[N],r[N];int cnt=1,root=1;

void add(int &rt,int l,int r,int x,int y,int w){

int len=r-l+1;

int l1=l^(w&(~(len-1)));// ~(len-1)代表求出类11000000的格式，再将w前面的几位屏蔽掉，在和L异或，就是他们的起点了。

int r1=l1+len-1;//l-r这个区间可以由l1-r1这个区间异或w得到

if(l1>=x&&r1<=y) return;

if(!rt) rt=++cnt;//动态开点

if(l1>y||r1<x){//将其他区间的值设为1；

tag[rt]=1;//标记该区间都是不合法的；

sum[rt]=len;

return;

}

if(tag[rt]) return;

int mid=(l+r)/2;

add(ch[rt][0],l,mid,x,y,w);

add(ch[rt][1],mid+1,r,x,y,w);

if(tag[rt]) sum[rt]=len;

else sum[rt]=sum[ch[rt][0]]+sum[ch[rt][1]];

}

void dfs(int u,int f,int w){

add(root,0,(1<<30)-1,l[u],r[u],w);

for(pii v:g[u]){

if(v.fi==f) continue;

dfs(v.fi,u,w^(v.sc));

}

}

void solve(){

cin>>n;

for(int i=1;i<=n;i++) cin>>l[i]>>r[i];

for(int i=1;i<n;i++){

int u,v,w;

cin>>u>>v>>w;

g[u].pb({v,w});

g[v].pb({u,w});

}

dfs(1,0,0);

cout<<(1<<30)-sum[1]<<"\n";

}

int main()

{

ios

int T=1;

while(T--)

{

solve();

}

return 0;

}