<https://www.cnblogs.com/SYCstudio/archive/2017/07/19/7206297.html>

思路：

对于将我们选定的数组中每次增加一个数字，这个数字的要求仅仅只是他存在某一位的二进制是1，并且前一个数字的对应位置上二进制也为1即可。

第一组循环找到当前可以取得的最大数字，然后通过新增加的这个数字来进行转移最大值，也就是第二个循环

#include<iostream>

#include<cstdio>

#include<cstdlib>

#include<cstring>

#include<algorithm>

typedef long long ll;

using namespace std;

int a[100010];

int f[100];

int main()

{

int n;

scanf("%d",&n);

for(int i=0;i<n;i++)

scanf("%d",&a[i]);

memset(f,0,sizeof(f));

for(int i=0;i<n;i++)

{

int now(0);

for(int j=0;j<=30;j++)

if((a[i]&(1<<j))!=0)

now=max(now,f[j]);

for(int j=0;j<=30;j++)

if((a[i]&(1<<j))!=0)

f[j]=max(now,f[j])+1;

}

int ans(0);

for(int i=0;i<30;i++)

ans=max(ans,f[i]);

printf("%d\n",ans);

return 0;

}