Analysis Of A Greedy Task

-Riya Sawhney

D. AND, OR and square sum

time limit per test: 2 seconds

memory limit per test: 512 megabytes

input: standard input

output: standard output

Gottfried learned about binary number representation. He then came up with this task and presented it to you.

You are given a collection of n non-negative integers $a_1, \ldots, a_n$. You are allowed to perform the following operation: choose two distinct indices $1 \le i, j \le n$. If before the operation $a_i = x$, $a_j = y$, then after the operation $a_i = x \text{ AND } y$, $a_j = x \text{ OR } y$, where AND and OR are bitwise AND and OR respectively (refer to the Notes section for formal description). The operation may be performed any number of times (possibly zero).

After all operations are done, compute $\sum_{i=1}^{n} a_i^2$ — the sum of squares of all a_i. What is the largest sum of squares you can achieve?

Input

The first line contains a single integer n ($1 \le n \le 2 \cdot 10^5$).

The second line contains n integers $a_1, \ldots, a_n$ ($0 \le a_i < 2^{20}$).

Output

Print a single integer — the largest possible sum of squares that can be achieved after several (possibly zero) operations.

Sample Cases:

1

123

$(123)^2 = 15129$

3

1 3 5

1 11 101

$1^2 + 3^2 + 5^2$

$= 35$

$$\left.\begin{array}{l} 1 \,\&\, 11 = 1 \\ 1 \mid 11 = 11 \end{array}\right\}$$

$$\left.\begin{array}{l} 1 \,\&\, 101 = 1 \\ 1 \mid 101 = 101 \end{array}\right\}$$

$$\left.\begin{array}{l} 11 \,\&\, 101 = 1 \\ 11 \mid 101 = 111 \end{array}\right\} \quad 1, 1, 7$$

$1^2 + 1^2 + 7^2$

$= 51$

$\boxed{1, 1, 7}$

Mutual undependence of bits

$a_i = \ldots\ldots 101 \ldots\ldots$ $\qquad$ $a_j = \ldots\ldots 011 \ldots\ldots$

$1 \& 1 = 1$ $\qquad$ $0 \& 0 = 0$ $\qquad$ $1 \& 0 = 0$

$1 \| 1 = 1$ $\qquad$ $0 | 0 = 0$ $\qquad$ $1 | 0 = 1$

⇒ number of on-bits occuring at a position remains constant

Shifts a particular bit from one number to the other

HOW MANY OPERATIONS?

x, y
$(x \geq y)$

$x^2 + y^2$

$x \& y = y - d$

$x | y = x + d$

$(x+d)^2 + (y-d)^2 = x^2 + y^2 + 2d^2 + 2d(x-y)$

unfinitely many?
(Not requured)

$\underline{d = 0}$?

$d \neq 0$ when y is on but x is off
$(x \geq y)$

$\Rightarrow d = 0$ $(x \geq y)$ x is off $\Rightarrow$ y is off (SUBMASK)

Count the number of on-bits at position 'i'

Reassign all the bits, higher priority to larger numbers

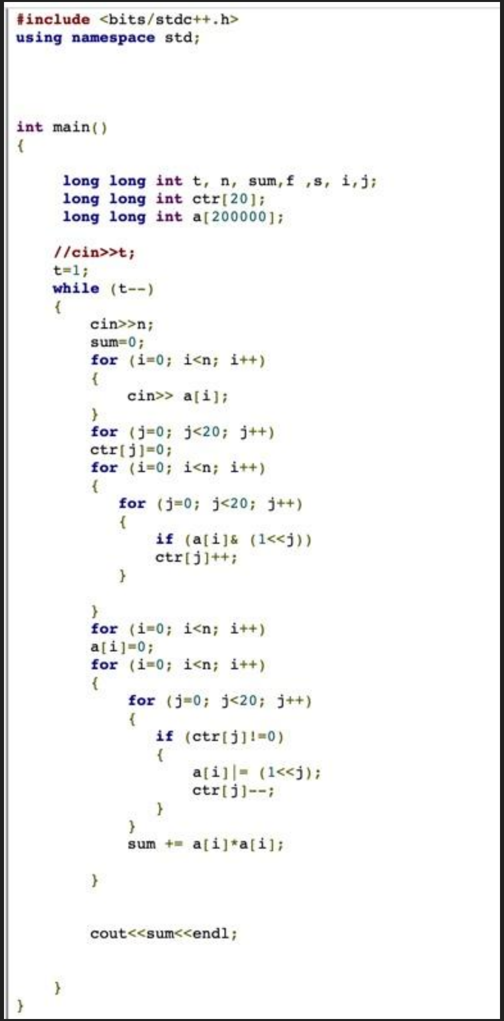
$n = 6$

$0 \le a_i \le 2^5$

$$\begin{array}{ccccc}
\bar{3} & \bar{1} & \bar{0} & \bar{2} & \bar{4} \\
\end{array}$$

$$\left\{\begin{array}{ccccc}
1 & 1 & 0 & 1 & 1 \\
1 & 0 & 0 & 1 & 1 \\
1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{array}\right\}$$

$$\begin{array}{ccccc}
1 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 1
\end{array}$$



```cpp
#include <bits/stdc++.h>
using namespace std;



int main()
{

     long long int t, n, sum,f ,s, i,j;
     long long int ctr[20];
     long long int a[200000];

    //cin>>t;
    t=1;
    while (t--)
    {
        cin>>n;
        sum=0;
        for (i=0; i<n; i++)
        {
            cin>> a[i];
        }
        for (j=0; j<20; j++)
        ctr[j]=0;
        for (i=0; i<n; i++)
        {
           for (j=0; j<20; j++)
           {
               if (a[i]& (1<<j))
               ctr[j]++;
           }

        }
        for (i=0; i<n; i++)
        a[i]=0;
        for (i=0; i<n; i++)
        {
            for (j=0; j<20; j++)
            {
               if (ctr[j]!=0)
               {
                   a[i]|= (1<<j);
                   ctr[j]--;
               }
            }
            sum += a[i]*a[i];

        }


        cout<<sum<<endl;


    }
}
```