Not So Random. (Google Apac University test - 2016)
https://code.google.com/codejam/contest/8264486/dashboard#s=p2

The smaller test case can be solved using naive method which is in O(3^N) where K is number of machines.Unfortunately the large test case needs dynamic approach.

This dynamic problem can be solved by observing the fact that when a bitwise operation is done between two constants (X and K) and their ouputs only a limited set of outputs are seen.
Example: Take X = 5 (101) and K = 3 (011)

Now observe this :
1. X ^ K = 110
2. X & K = 001
3. X | K = 111

4. X ^ K ^ K = 101 (repeated)
5.X & K ^ K = 010
6. X | K ^ K = 100

7. X ^ K & K = 010 (repeated)
8. X & K & K = 001 (repeate)
9. X | K & K = 011

10. X ^ K | K = 111 (repeated)
11. X & K | K = 011 (repeated)
12. X | K | K = 111 (repeated)

Following this pattern , after 12 . everything you get will be repeated.Interestingly I found out that given any values of X and K , one will not get more than 6 unique results (Though I failed to understand why).

After understanding this sample test case you will be able to understand the code:

Input :
1
2 5 3 10 40 50

Output :
5.02

initial input is X = 101
results after the input passes through first machine:
probabilty of getting X & K = 001 is 0.1
probabilty of getting X | K = 111 is 0.4
probabilty of getting X ^ K = 110 is 0.5

Now these input will pass through the second machine and the probabilties are the following:

probabilty of getting (X & K) & K = 001 is 0.1*0.1 = 0.01 (Probabilty rule of multiplication –
getting “&” in both the first and second machine)

probabilty of getting (X & K) | K = 011 is 0. 1*0.4 =0.04

probabilty of getting (X & K) ^ K = 010 is 0.1*0.5 = 0.05

Probability of getting (X | K) & K = 011 is 0.4 * 0.1 =0.04
And So on......
We get :
(001)*0.01 + (011)*(0.04+0.04) +(010)*(0.05+0.05) + (111)*(0.16+0.2) + (100)*(0.2) + (101)*(0.25) = 5.02

Basic idea : You can store these values in a HashMap where key is the Integer and corresponding value as the probability of that Intger to occur.