Distinct Powers - Writeup

This exercise is marked as the challenge for VG101(SU2020) - Lab 2.

Problem Statement

Consider all integer combinations of a^b for $2 \le a \le 5$ and $2 \le b \le 5$:

$$\begin{gathered}2^2 = 4, 2^3 = 8, 2^4 = 16, 2^5 = 32 \\ 3^2 = 9, 3^3 = 27, 3^4 = 81, 3^5 = 243 \\ 4^2 = 16, 4^3 = 64, 4^4 = 256, 4^5 = 1024 \\ 5^2 = 25, 5^3 = 125, 5^4 = 625, 5^5 = 3125\end{gathered}$$

If they are lined up, with any repeats removed, we get the following sequence of 15 distinct terms:

$4, 8, 9, 16, 25, 27, 32, 64, 81, 125, 243, 256, 625, 1024, 3125$

How many distinct terms are in the sequence generated by a^b for

- 2 ≤ a ≤ 10 and 2 ≤ b ≤ 10? (0.05 marks)
- 2 ≤ a ≤ 100 and 2 ≤ b ≤ 100? (0.05 marks)
- 2 ≤ a ≤ 100000 and 2 ≤ b ≤ 100000? (0.05 marks)

Note: try to improve the efficiency as possible as you can!

Writeups collected from VG101-SU2020

Scarlet | 0.367s | C++

Consider all the number $z_i (2 \le z_i \le 10^5)$ that can't be represented as x^y $(y > 1)$. They are

$2, 3, 5, 6, 7, 10, \ldots$

We just need to sum up the number of distinct z_i^k.

Valid k can be found in the $\lfloor \log n \rfloor \times n$ table, in $n = 10^5$ this case, they are

$1, 2, 3, \cdots 100000$
$2, 4, 6, \cdots 200000$
$3, 6, 9, \cdots 300000$
$\cdots$

For each z_i, we can find the greatest t s.t. $z_i^t < 10^5$, we just want to know how many distinct numbers are there in the first t lines of the above table. Calculate this number and add them up for each z_i, you will get the answer.

No available code in MATLAB :D (See lc's code)

Author: Muchen Xu (TA)

limlimg | 0.011405s | MATLAB

Solution is described in the comments of the code.

```
% revised chanllenge code
% time complexity O(log2(a)*(b + a^0.5))
% run time for a = 100000, b = 100000 case: < 0.1"
a = input("");
b = input("");
tic;

% For ai < aj in 2:a, only when aj = ai^n will their results of a^b
% collide. So two numbers are grouped together if one of them is the power
% of the other.

% The number of powers generated by a whole group is only dependent on its
% size. This mapping is provided by calcGroupValue beforehand.
GroupValue = calcGroupValue(a, b);

% Firstly, put every number in individual groups. If a number is later
% found to be in another group marked by a smaller number, subtract the
% result then.
result = (int64(a)-1)*(int64(b)-1);
GroupRecord = zeros(fix(sqrt(a)), 1, "int8");
for i = 2:size(GroupRecord, 1)
% i above sqrt(a) are not checked because they either form their own groups
% with no other number or belong to a group with a number within sqrt(a).

    % If i doesn't belong to any group
    if GroupRecord(i)
        continue;
    end
    % then it must be the smallest number of its own group.
    % Count the size of this group.
    PowerOfI = i;
    for j = 1:fix(log2(a))
        if(PowerOfI <= size(GroupRecord, 1))
            GroupRecord(PowerOfI) = i;
        end
        PowerOfI = PowerOfI * i;
        if PowerOfI > a
            break;
        end
    end
    % All numbers in this group were assumed to generate b-1 powers before.
    % Subtract the extra number.
    result = result - j*GroupValue(1) + GroupValue(j);
end
disp(result);
toc;

function result = calcGroupValue(a, b)
% Calculate the number of powers generated by a group containing 1 ... n
% members and return the result in a vector.

% The values are only dependent on b and the biggest size of groups is
% determined by a, so once the inputs are given the results can be
% generated for repeated use.

    MaxSize = fix(log2(a));
    result = zeros(MaxSize, 1);
    record = zeros(MaxSize * b, 1, "int8");
    for i = 1:MaxSize
    % Construct a group in the order of a^1, a^2 ...
    % Each member added results in the addition of the powers generated by
    % this group, so result(n) = result(n-1) + f(n), where f(n) is the
    % number of distinct powers generated by a^n.
        if i > 1
            result(i) = result(i-1);
        end
        for j = 2:b
            if record(i*j) == 0
                result(i) = result(i) + 1;
                record(i*j) = 1;
            end
        end
    end
end
```

Author: Yifan Shen

lc | 0.018473s | MATLAB

This program works in the way which is described by Muchen Xu.

```
clearvars,clc;
num=input('n: ');
tic;
list=ones(1,num,'uint8');
su=zeros(1,num,'uint32');
cnt=zeros(1,num*20,'uint8');
record=zeros(20,'uint32');
for i=1:log2(num)
    cnt(int32(i)*int32((2:num)))=1;
    record(i)=sum(cnt,'all');
end
for i=2:num
    if list(i)>0
        t=2;
        while true
            if power(i,t)<=num
                list(power(i,t))=0;
                list(i)=list(i)+1;
                t=t+1;
            else
                break;
            end
        end
        su(i)=record(list(i));
    end
end
```

```
int64(sum(su,'all'))
toc;
```

Author: Chang Liu

Rip-Van-Winkle | 0.051941s | MATLAB

There still exist some places to improve. It is similar to what is described by Muchen Xu.

```
n = input('');
count = 0;
k = 0;
mark = zeros(n,1);
mark2 = zeros(floor(log2(n)),1);
tic
for i = 2 : n
    k = 0;
    if  mark(i) == 0
       while i ^ (k+1) <= n
           k = k + 1;
           mark(i ^(k - 1) , 1 ) = 1;
       end
       mark(i ^ k , 1 ) = 1;
       if mark2(k) ~= 0
           count = count + mark2(k);
       else
           hash = zeros(k * n,1);
           inner_cnt = 0;
           for j = 1 : k
               for a = 2 : n
                   if hash(a * j , 1) == 0
                       hash(a * j , 1) = 1;
                       count = count + 1;
                       inner_cnt = inner_cnt + 1;
                   end
               end
           end
           mark2(k) = inner_cnt;
       end
    end
end
fprintf('%.0f',count)
toc
```

Author: Runqing Cai

ISTHATDISTANCE | 0.193364s | MATLAB

The solution is same as above. (but constant may be larger)

```
clearvars,clc;
n=input("");
tic;
maxPower=floor(log2(n));
isRepeat=zeros(maxPower,n);
```

```
% (i,j) represent for a^i, a^(i*j) can be repeated or not
represent=zeros(n,1);
flag=zeros(n,1);

for i=2:n
    if (flag(i))
        continue;
    else
        for j=2:floor(log(n)/log(i))
            flag(i^j)=1;
        end
    end
end
for i=2:maxPower
    for j=1:i-1
        t=lcm(i,j);
        k=1;
        while (k*t/j<=n)
            isRepeat(i,k*t/i)=1;
            k=k+1;
        end
    end
end
represent(1)=n-1;
for i=2:maxPower
    represent(i)=represent(i-1);
    for k=2:n
        represent(i)=represent(i)+1-isRepeat(i,k);
    end
end
result=int64(0);
for i=2:n
    if flag(i)
        continue;
    else
        result=result+represent(floor(log(n)/log(i)));
    end
end
disp(result);
toc;
```

Author: Tao Lu