Project 1

Abraham Jacob Reines

March 27, 2022

1 Problem 1

Problem: Write an algorithm called factors that takes a positive integer n and outputs the number of factors for n.

1.1 Solution:

Matlab Code

```
function count=factors(n)
count=0;
for i=1:floor(sqrt(n))
    if mod(n,i)==0
        count=count+1;
        if n/i~=i
            count=count+1;
        end
    end
end
```

1.2 Description of solution:

The first if-statement sets a ceiling (floor) for the iteration to stop at the square of the positive integer. The second if-statement applies a mod function as discussed in class. The third if-statement checks the other number, for example, when dividing $42/6 = 7$, we are checking if 42 is not a square root and 2 different numbers.

1.3 Results and Conclusions:

All results from factors inputs produced reasonable factors! This indicates the algorithm is performing within constraints, such as positive integer inputs. Factors(10) = 4.

2 Problem 2

Problem: Which number less than five hundred thousand has the most factors? Write a loop that calls factors.

2.1 Description of solution:

An if-statement with 3 variables utilized in conjunction with factors from Problem 1 produces a number less than 500,000 with the most factors possible. The if-statement sets a ceiling for 500000 and uses the function to find the integer with the most factors.

2.2 Results and Conclusions:

All results from if-statement inputs produced reasonable output responses! This indicates the algorithm is performing properly for the problem. Answer: the number less than 500,000 with the most factors is 498,960.

2.3 Solution:

Matlab Code

```
n=0;r=0;q=0;
for i=1:500000
    n=factors(i);
    if n>=r
        r=n;
        q=i;
    end
end
disp('Desired number is: '), disp(q)
```

3 Problem 3

Problem: Which number less than five hundred thousand has the most factors? Write a loop that calls factors.

3.1 Solution:

Matlab Code

```
function Output=prime(n)
nf=0;
Output=0;
nf=factors(n);
if nf>1 && nf<3
    Output=1;
else
    Output=0;
end
end
```

```
function yorn = prime2(n)
count=0;
for i=1:floor(sqrt(n))
     if mod (n,i)==0
         count=count+1
         if n/i ~= i
             count=count+1;
         end
     end
end
if count>2
    yorn=1
else
    yorn=0
end
```

3.2 Description of solution:

Two different functions solve this problem. One utilizes a combination of if and else-statements. The if-statement is crucial to obtaining the prime from factors. The bottom algorithm is a function more similar to Problem 1.

3.3 Results and Conclusions:

All results from prime function inputs produced accurate output responses! This indicates the algorithm is performing properly for the problem constraints.

4 Problem 4

Problem: Using the function from Problem 3, what is the 6000th prime?

4.1 Description of solution:

Utilizing the prime function from Problem 3, add a while loop to find the 6000th prime

4.2 Results and Conclusions:

All results from prime function inputs produced accurate output responses! This indicates the algorithm is performing properly for the problem constraints. The 6000th prime is 59,359.

4.3 Solution:

```
function yorn = prime3(n)
count=0;
for i=1:floor(sqrt(n))
     if mod (n,i)==0
         count=count+1
         if n/i ~= i
             count=count+1;
         end
     end
end


if count>2
    yorn=1
else
    yorn=0
end
end

if count>2
    yorn=1;
else
    yorn=0;
end

end

i=0; j=2
while i<6000
    if prime(j)==0
        i++;
    end
    j++;
end
disp(j-1)
```

5 Problem 5

Problem: Write a Matlab routine called 'listprimes' that takes as input a positive integer n and outputs a one dimensional array where the ith element is one if i is prime, zero if it is not. Use this function to verify the 6000th prime from question four above. Comment on relative speed.

5.1 Solution:

```
clear all
clc
n=60000;a=1:60000;
[prime,output]=listprime(n);
fprintf('n=%d\n',n)
z=find(output>0);
fprintf('6000th prime is %d\n',a(z(6000)))

function [prime,output]=listprime(n)

a=1:n;
i=2;

while i<=sqrt(n)
    b=i+i:i:n;
    for j=1:length(b)
        a(b(j))=0;
    end
    C=find(a~=0);
    d=find(C>i);
    i=C(d(1));
end

a(1)=0;
prime=a;
prime(prime==0)=[];
output=a;
output(find(output>0))=1;
end
```

```
clear all
clc
n=60000;a=1:60000;
[prime,output]=listprime(n);
fprintf('n=%d\n',n)
z=find(output>0);
fprintf('6000th prime is %d\n',a(z(6000)))

function [prime,output]=listprime(n)

a=1:n;
i=2;

while i<=sqrt(n)
    b=i+i:i:n;
    for j=1:length(b)
        a(b(j))=0;
    end
    C=find(a~=0);
    d=find(C>i);
    i=C(d(1));
end

a(1)=0;
prime=a;
prime(prime==0)=[];
output=a;
output(find(output>0))=1;
end
```

n=60000
6000th prime is 59359

Published with MATLAB® R2021b

5.2 Description of solution:

This problem requires two parts, the first finds primes up to n, the second finds the prime for input value.

5.3 Results and Conclusions:

All results from prime function inputs produced reasonable output responses! This indicates the algorithm is performing properly for the problem. The computation is relatively quick. Answer: the 6000th prime is 59,359.

6 Problem 6

Problem: Write a Matlab routine called 'listprimes' that takes as input a positive integer n and outputs a one dimensional array where the ith element is one if i is prime, zero if it is not. Use this function to verify the 6000th prime from question four above. Comment on relative speed.

6.1 Solution:

Compared to Problem 2, how can we add loops to utilize factors more effectively? How much faster is this method and how can we find the most divisors beyond five hundred thousand?

```
clear all
clc

n=500000;
z=factors(n);
[mx,index]=max(z);
fprintf('n=%d\n',n)
fprintf('Number of factors for %d ',mx)
fprintf('Number of is %d\n ',index)

function z=factors(n)
z=ones(n,1);
for i=2:n
     for j=i:i:n
          z(j)=z(j)+1;
     end
end
end
```

```
clear all
clc

n=500000;
z=factors(n);
[mx,index]=max(z);
fprintf('n=%d\n',n)
fprintf('Number of factors for %d ',mx)
fprintf('Number of is %d\n ',index)

function z=factors(n)
z=ones(n,1);
for i=2:n
    for j=i:i:n
        z(j)=z(j)+1;
    end
end
end
```

```
n=500000
Number of factors for 200 Number of is 498960
```

Published with MATLAB® R2021b

6.2 Description of solution:

This problem requires an algorithm that utilizes factors function and two for loops. The loops find factors up to n.

6.3 Results and Conclusions:

All results from factors function inputs produced reasonable output responses! This indicates the algorithm is performing properly for the problem. The elapsed time for this method is much faster! By several magnitudes. Answer: Maximum factors can be observed for 200 with number of factors 498,960.