Count prime

Count prime

- Given an integer n, return the number of prime numbers that are strictly less than n.
- Input: n = 10
- Output: 4
- Explanation: There are 4 prime numbers less than 10, they are 2, 3, 5, 7.

10 points out of a total 25 points that your program can run n<10^4

- iterated from i = 2 to i<n
  - iterated from j = 2 to j<i, check that whether i is perfectly divisible by j
    - If i is perfectly divisible by j, i is not a prime number, then terminated the loop
  - Otherwise, i is a prime number.

15 points out of a total 25 points that your program can run 10^5<=n<2.5*10^6 in one second.

- iterated from i = 2 to i<n
  - iterated from j = 2 to j*j<=i, check that whether n is perfectly divisible by j
    - If i is perfectly divisible by j, i is not a prime number, then terminated the loop
  - Otherwise, i is a prime number.

若N為合數，且有r個因數$(r \geq 2)$

反證明 $P_1, P_2, \cdots, P_r > \sqrt{N}$

$$N = P_1 \times P_2 \times \cdots \times P_r > N^{\frac{r}{2}}$$

$$N^{\frac{r}{2}} \geq N \ (\because \frac{r}{2} \geq 1)$$

$\because N > N^{\frac{r}{2}}$ 和 $N^{\frac{r}{2}} \geq N$ 矛盾

$\therefore$ 反證成立

If it is a composite number, it must have at least one factor<=$\sqrt{n}$

25 points out of a total 25 points that your program can run 3.3*10^6<=n<=5*10^6 in one second.

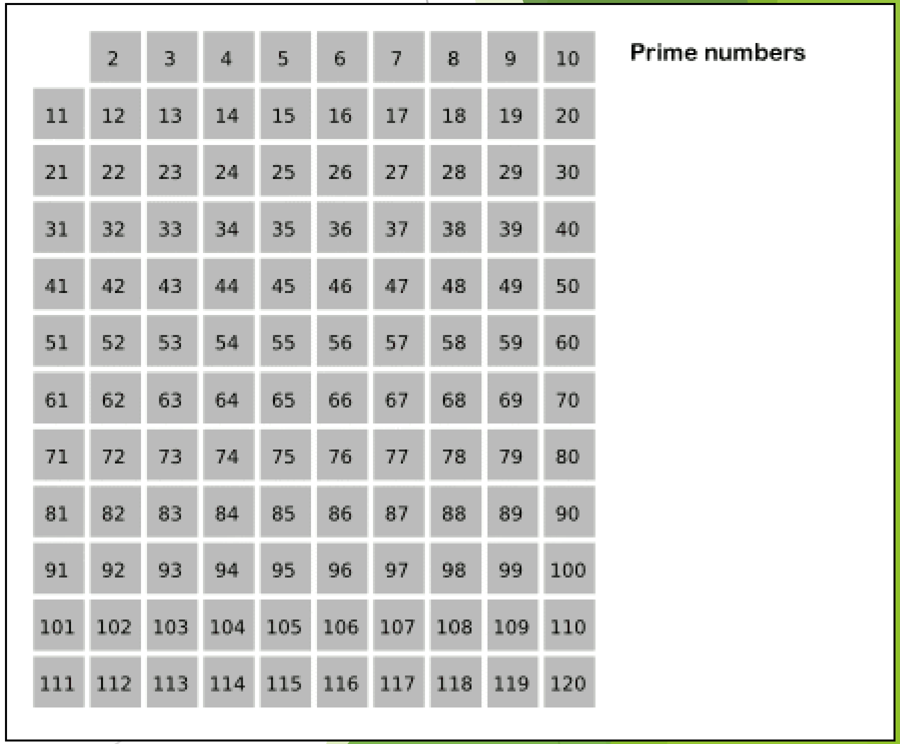
Sieve of Eratosthenes

algorithm Sieve of Eratosthenes is
    input: an integer n > 1.
    output: all prime numbers count from 2 through n-1.

    let A be an array of Boolean values, indexed by integers 2 to n-1, initially all set to true.

    for i = 2, 3, 4, ..., not exceeding $\sqrt{n}$ do
        if A[i] is true
            for j = i^2, i^2 +i, i^2 +2i, i^2 +3i, ..., not exceeding n do
                A[j] := false

    return all i such that A[i] is true.



Questions?