**RECITATION 9: LIST CLIENT**

1. WHAT IS LIST?

An ordered collection of items of some element type E. Note that this doesn't mean that the objects are in ***sorted*** order, it just means that each object has a ***position*** in the List, starting with position zero.

A list is a dynamic ordered tuple of homogeneous elements A0, A1, A2, …, AN-1

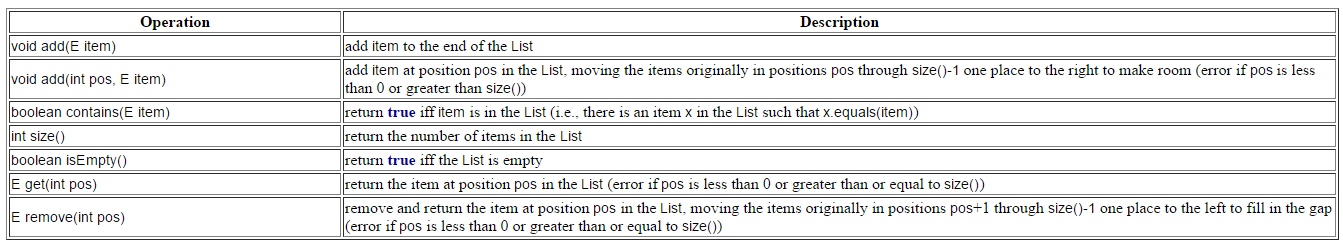
where Ai is the i-th element of the list.

The position of element Ai is i;

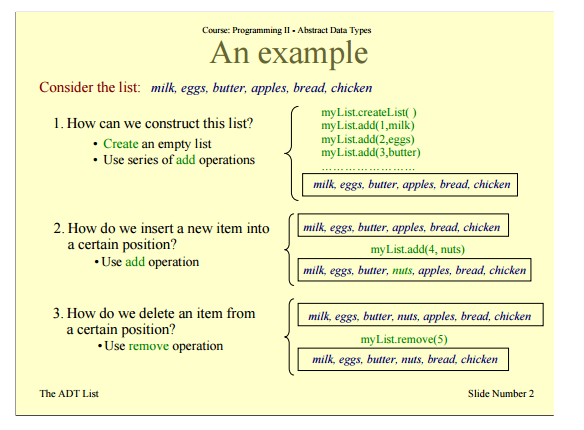
positions range from 0 to N-1 inclusive!

The size of a list is N (a list with no elements is called an “empty list”)

1. GENERIC OPERATIONS:



1. EXAMPLE:



1. SIEVE OF ERATOSTHENES:

A prime number is a natural number that has exactly two distinct natural number divisors: 1 and itself.

To find all the prime numbers less than or equal to a given integer n by Eratosthenes' method:

Create a list of consecutive integers from 2 through n: (2, 3, 4, ..., n).

Initially, let p equal 2, the smallest prime number.

Enumerate the multiples of p by counting to n from 2p in increments of p, and mark them in the list (these will be 2p, 3p, 4p, ... ; the p itself should not be marked).

Find the first number greater than p in the list that is not marked. If there was no such number, stop. Otherwise, let p now equal this new number (which is the next prime), and repeat from step 3.

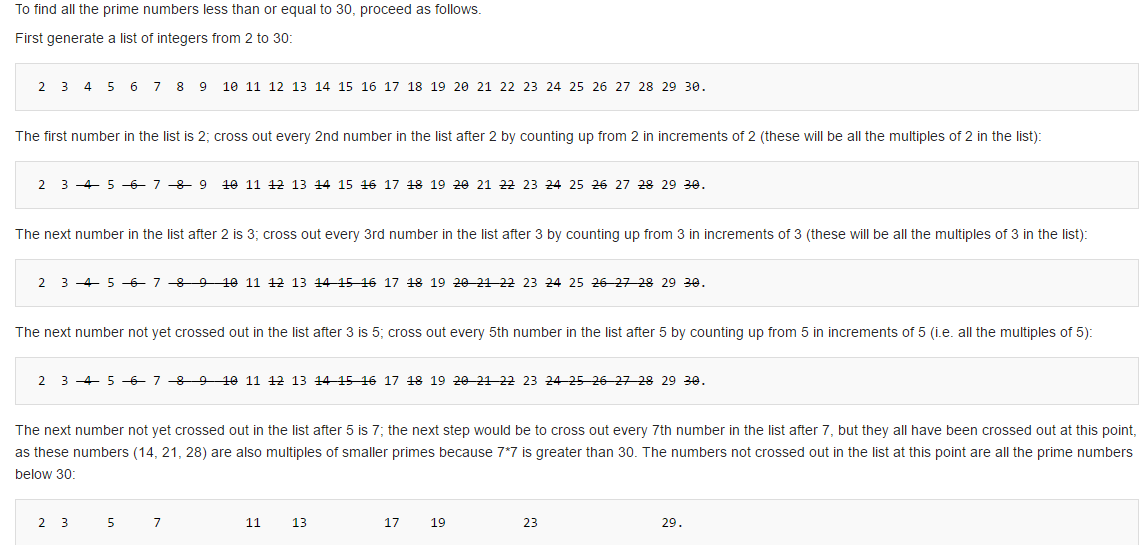
When the algorithm terminates, the numbers remaining not marked in the list are all the primes below n.

The main idea here is that every value given to p will be prime, because we have already marked all the multiples of the numbers less than p. Note that some of the numbers being marked may have already been marked earlier (e.g., 15 will be marked both for 3 and 5).

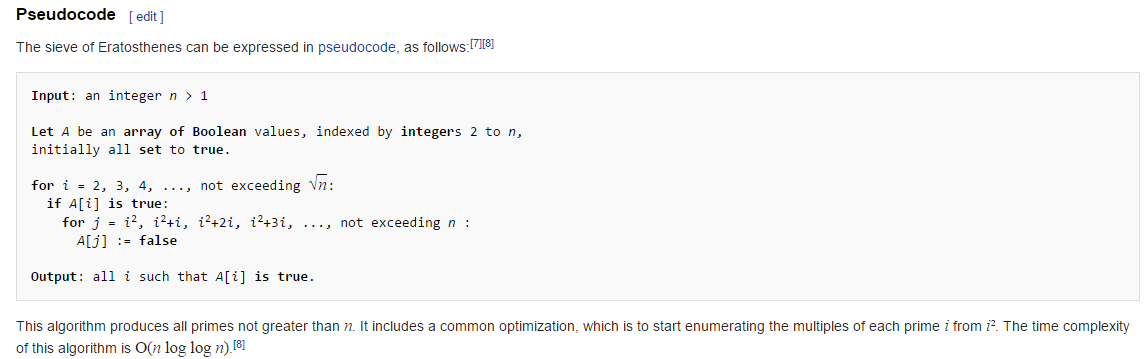
As a refinement, it is sufficient to mark the numbers in step 3 starting from p2, as all the smaller multiples of p will have already been marked at that point. This means that the algorithm is allowed to terminate in step 4 when p2 is greater than n.

Another refinement is to initially list odd numbers only, (3, 5, ..., n), and count in increments of 2p from p2 in step 3, thus marking only odd multiples of p. This actually appears in the original algorithm. This can be generalized with wheel factorization, forming the initial list only from numbers coprime with the first few primes and not just from odds (i.e., numbers coprime with 2), and counting in the correspondingly adjusted increments so that only such multiples of p are generated that are coprime with those small primes, in the first place.

EXAMPLE :



1. ALGORITHM:



1. EXERCISE:

* Using the Sieve of Eratosthenes, determine the prime numbers present in a particular range of numbers. (Ex: Find the prime number between 1 to 10)
* Modify the Sieve.java: Change the primesUpTo method in Sieve.java such that given the max range value ; the method returns list containing valid prime numbers in that range.