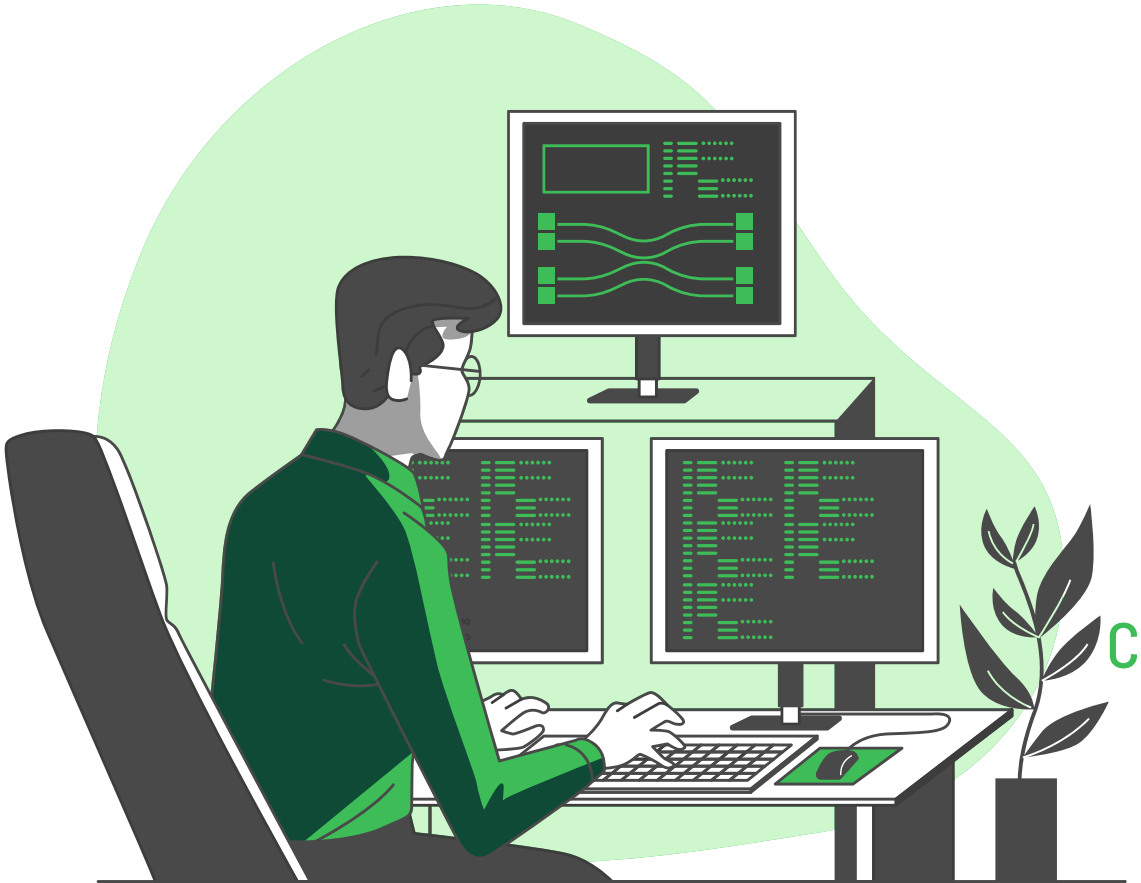


Course Overview

CS 251 - Data Structures and Algorithms

CS 251 – Data Structures and Algorithms

01

Data Structure

Manage data in memory and define support operations.

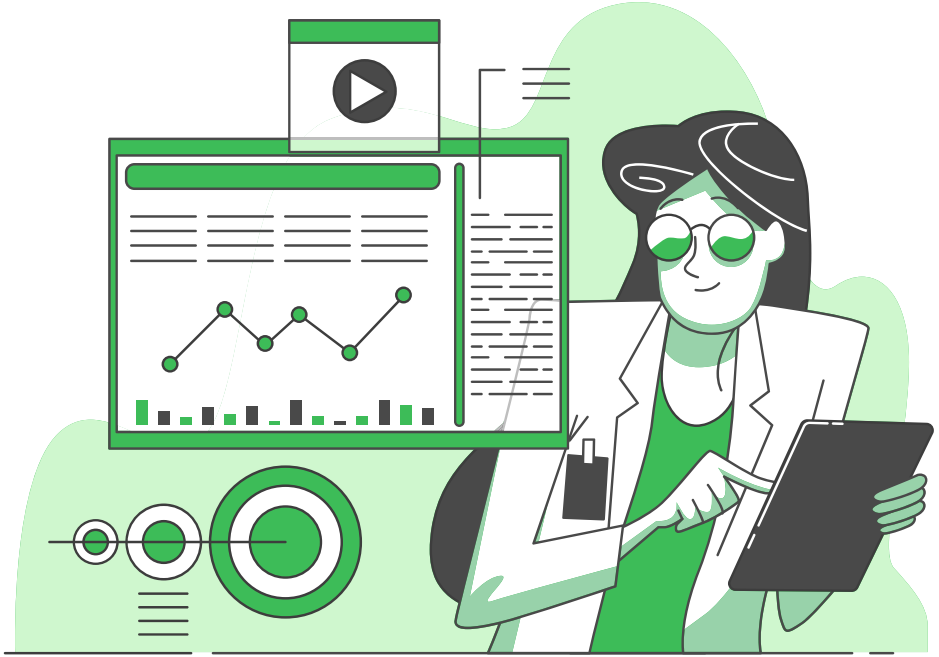
02

Algorithm

Computational steps that receive input and produce output.

These are extremely simplified definitions. We will cover them properly in the upcoming sessions.

Table of Contents



01

Finding Prime Numbers

A basic warm-up example

Prime Numbers

"A Prime Number is a natural number greater than 1 that is not product of two smaller natural numbers."

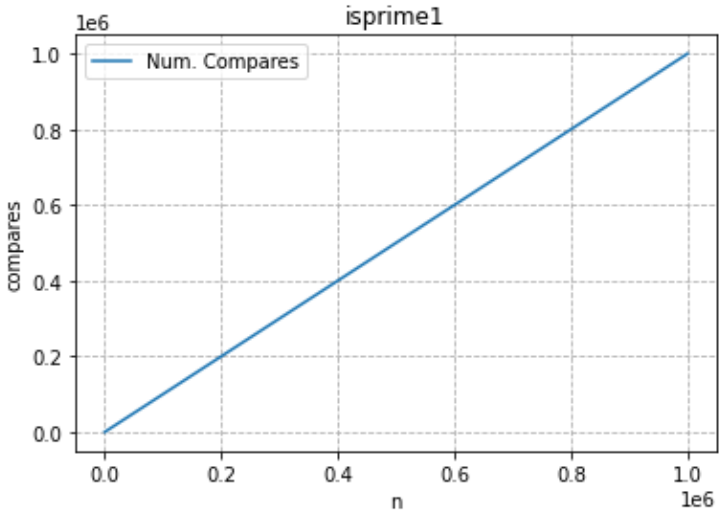
- Wikipedia -

Primality Test

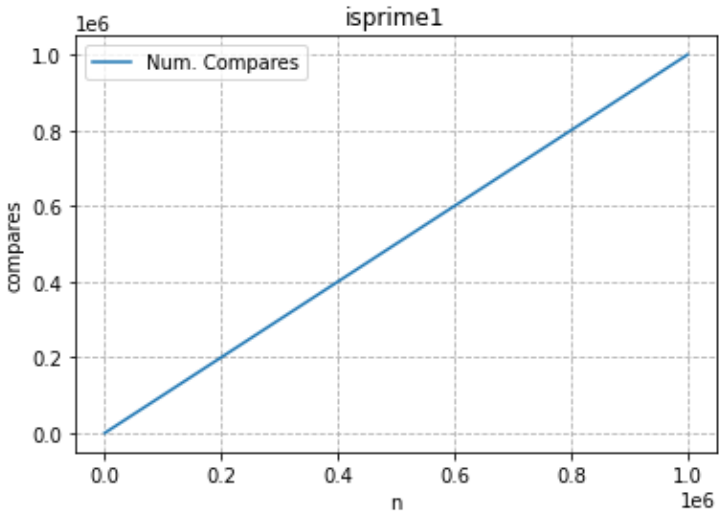
"A Prime Number is a natural number greater than 1 that is not product of two smaller natural numbers."

```
algorithm isprime1(n:integer) → boolean

   if n <= 1 then
      return false
   end if

   for i from 2 to n-1 do
      if n mod i = 0 then
         return false
      end if
   end for

   return true

end algorithm
```

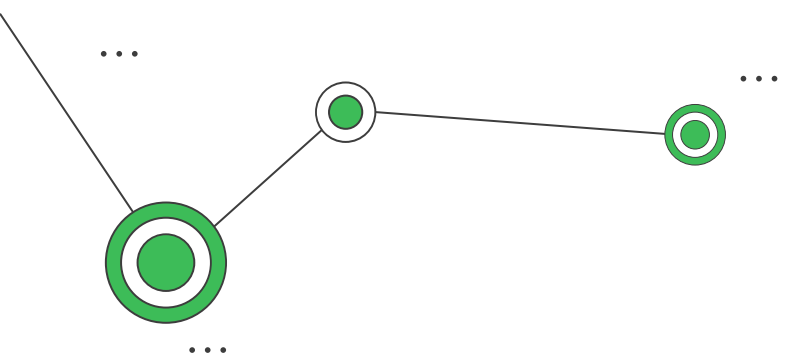
```
algorithm isprime1(n:integer) → boolean

   if n <= 1 then
      return false
   end if

   for i from 2 to n-1 do
      if n mod i = 0 then
         return false
      end if
   end for

   return true

end algorithm
```



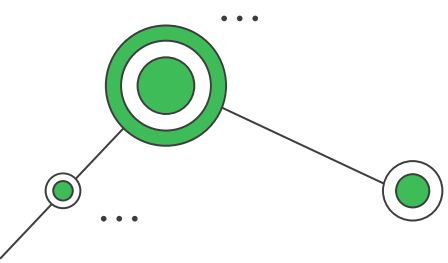
```java
public boolean isprime1(int n)
{
   if (n <= 1)
   {
      return false;
   }

   for (int i = 2; i < n; i += 1)
   {
      if (n % i == 0)
      {
         return false;
      }
   }

   return true;
}
```



Can we do better?



Compare until $\sqrt{n}$



If a number n is **not a prime**, it can be factored into **two factors** $a, b \in \mathbb{N}$ such that:

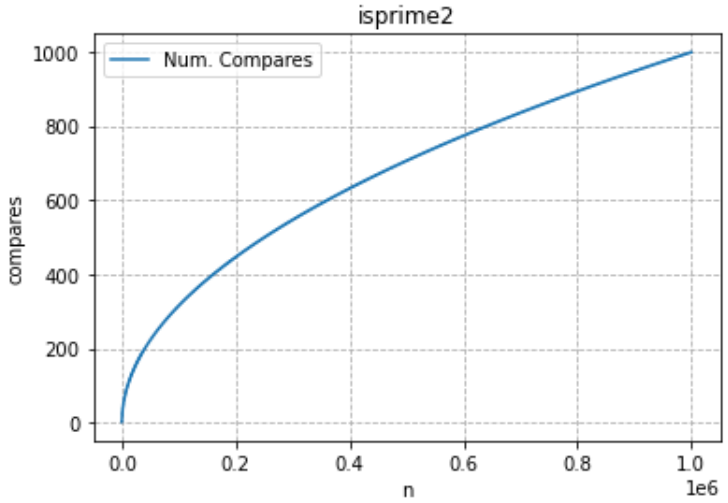
$$n = ab$$

Notice that a and b can not be both greater than $\sqrt{n}$, since ab would be greater than $\sqrt{n}\sqrt{n} = n$.

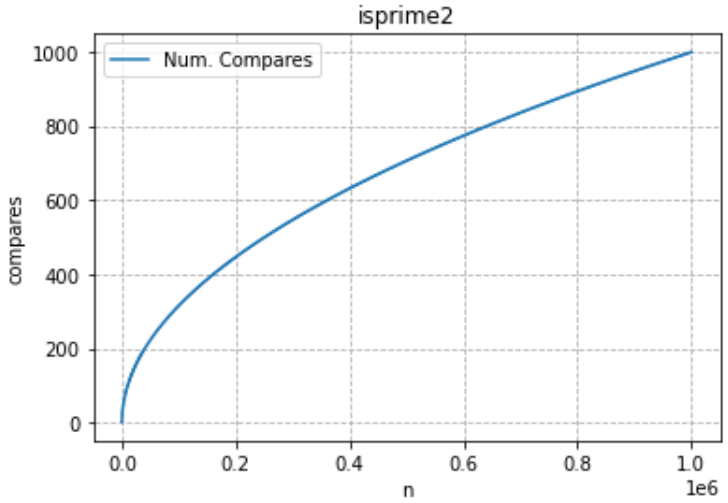
So, in any factorization of n, at least one of the factors must be smaller than the square root of n, and if we can not find any factors less than or equal to $\sqrt{n}$, then n must be a prime.

- Adapted from Stack Overflow -

```
algorithm isprime2(n:integer) → boolean

   if n <= 1 then
      return false
   end if

   for i from 2 to sqrt(n) do
      if n mod i = 0 then
         return false
      end if
   end for

   return true

end algorithm
```

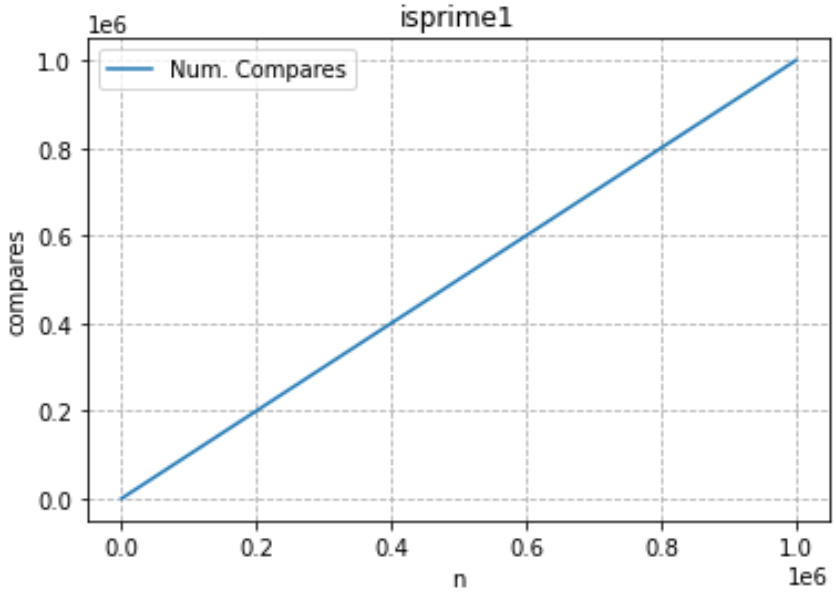


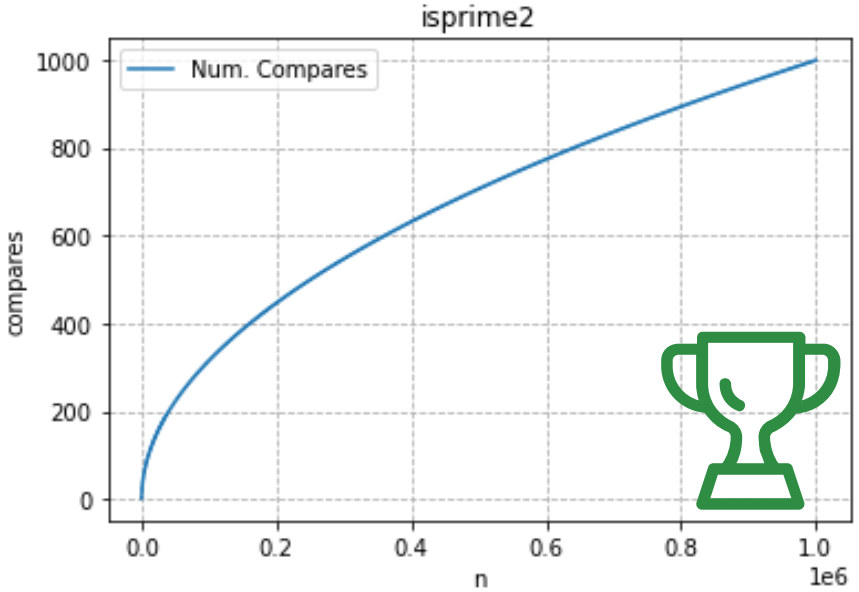
```java
public boolean isprime2(int n)
{
   if (n <= 1)
   {
      return false;
   }

   for (int i = 2; i * i <= n; i += 1)
   {
      if (n % i == 0)
      {
         return false;
      }
   }

   return true;
}
```

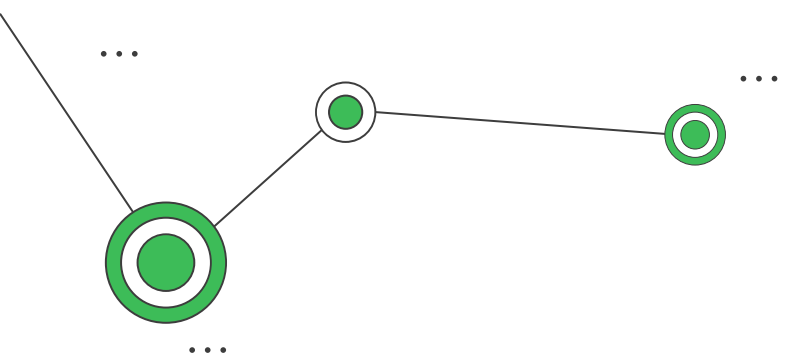


Can we do better?

isprime1 –VS– isprime2







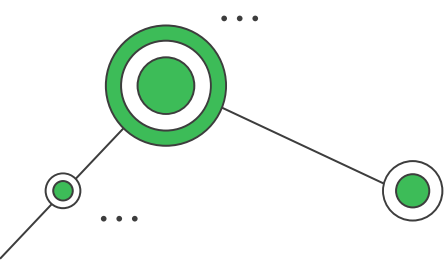
Numbers of the form $6k \pm 1$

All integers can be expressed as $6k + i$, where $k \in \mathbb{Z}$ and $i = -1, 0, 1, 2, 3, 4$.

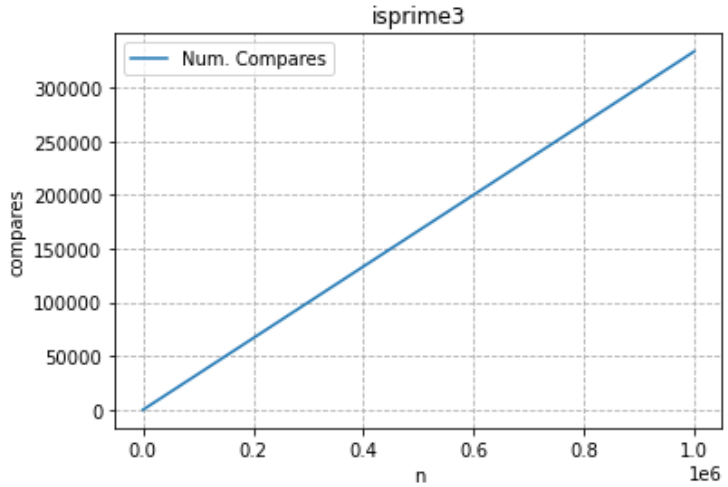
Prime numbers greater than 3 are of the form $6k \pm 1$, for $k \in \mathbb{Z}^+$. Do you see why?

$6k - 1$: Odd. Could be prime or not.
$6k + 0$: Even.
$6k + 1$: Odd. Could be prime or not.
$6k + 2$: Even.
$6k + 3$. Odd. Multiple of three.
$6k + 4$: Even.

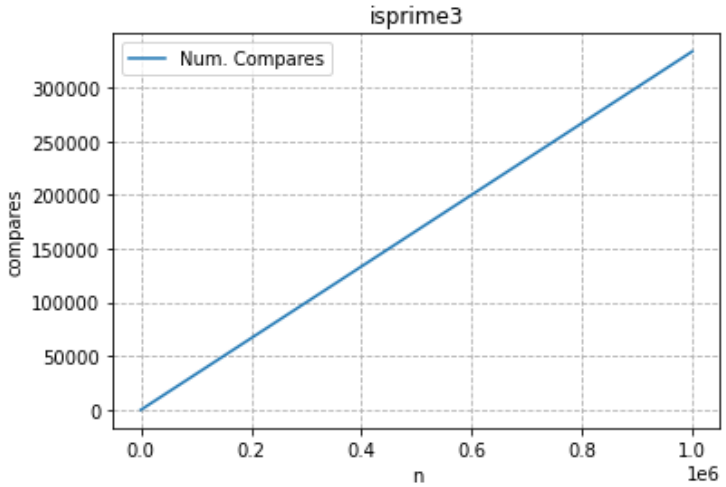
Question: Is $6k \pm 1$ always a prime number?



```
algorithm isprime3(n:integer) → boolean

   if n <= 1 then
      return false
   end if

   if n <= 3 then
      return true
   end if

   if n mod 2 = 0 or n mod 3 = 0 then
      return false
   end if

   for i from 5 to n-2 step 6 do
      if n mod i = 0 or n mod (i + 2) = 0 then
         return false
      end if
   end for

   return true

end algorithm
```

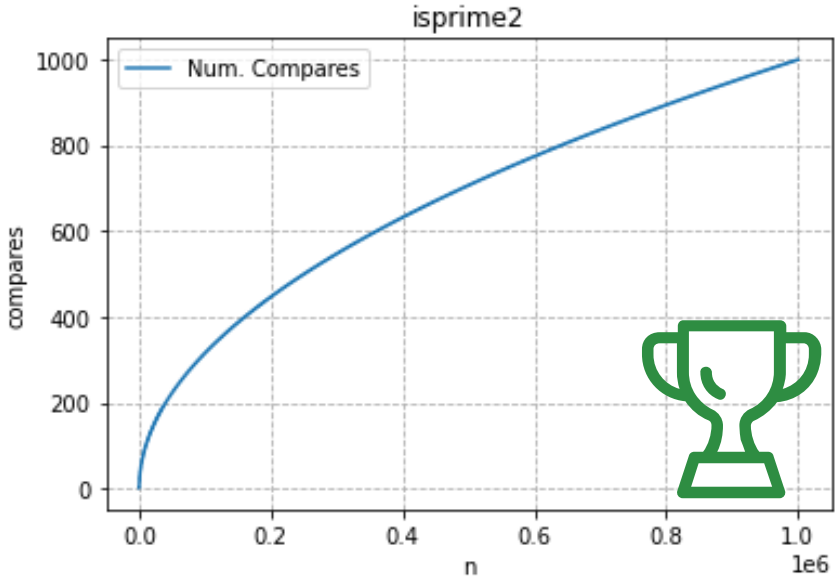


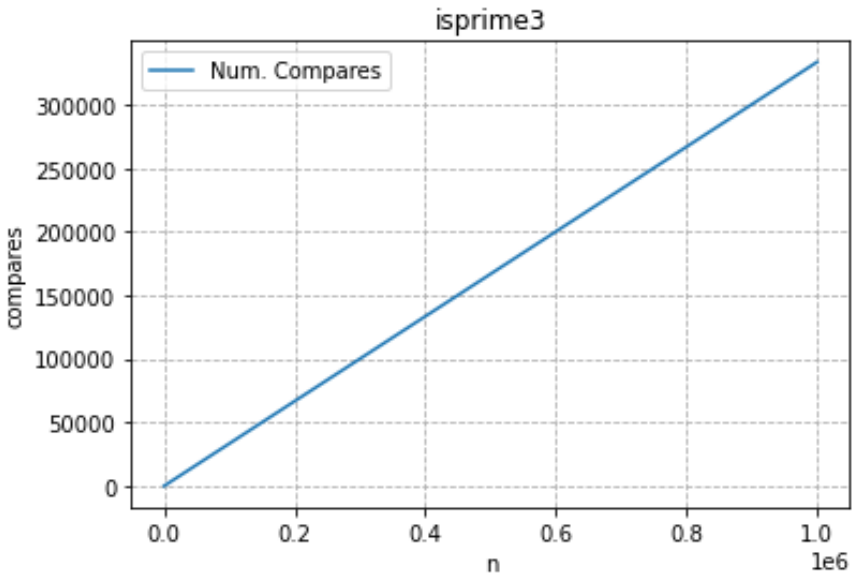
```java
public boolean isprime3(int n)
{
   if (n <= 1)
   {
      return false;
   }

   if (n <= 3)
   {
      return true;
   }

   if (n % 2 == 0 || n % 3 == 0)
   {
      return false;
   }

   for (int i = 5; i < n - 2; i += 6)
   {
      if (n % i == 0 || n % (i + 2) == 0)
      {
         return false;
      }
   }

   return true;
}
```



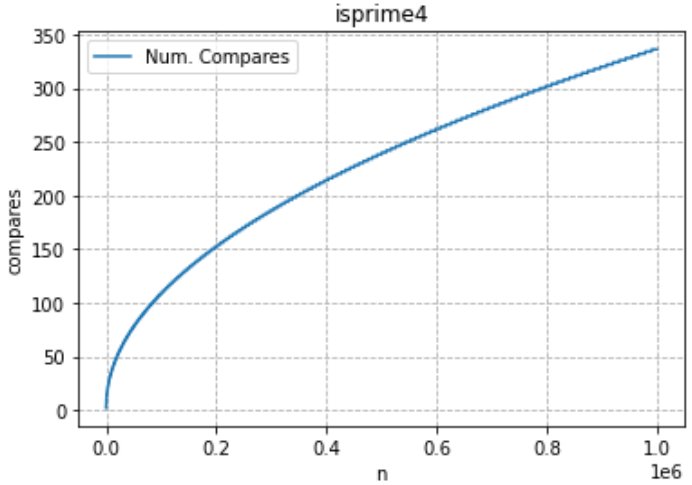
Can we do better?

isprime2 –VS– isprime3

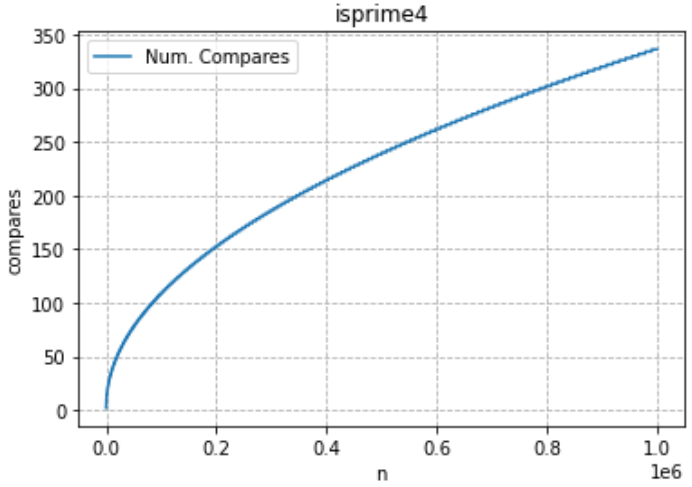




```
algorithm isprime4(n:integer) → boolean

   if n <= 1 then
      return false
   end if

   if n <= 3 then
      return true
   end if

   if n mod 2 = 0 or n mod 3 = 0 then
      return false
   end if

   for i from 5 to sqrt(n) step 6 do
      if n mod i = 0 or n mod (i + 2) = 0 then
         return false
      end if
   end for

   return true

end algorithm
```

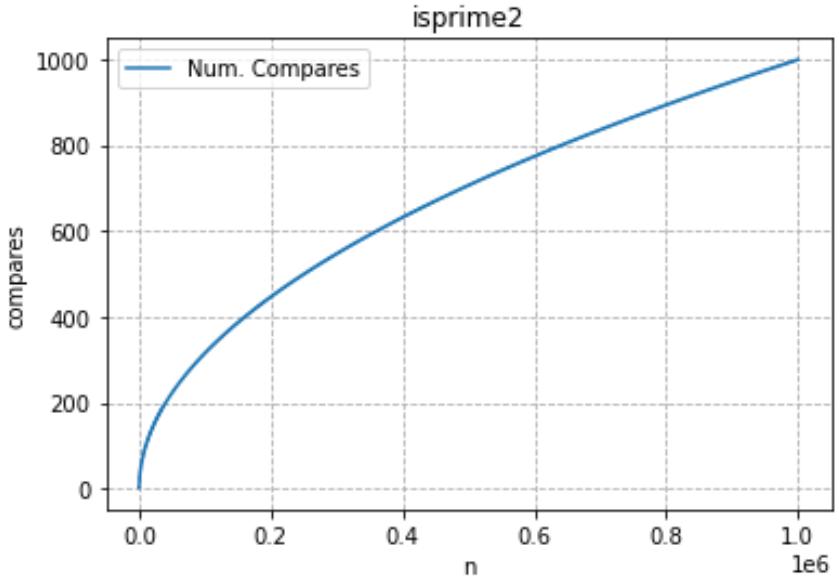


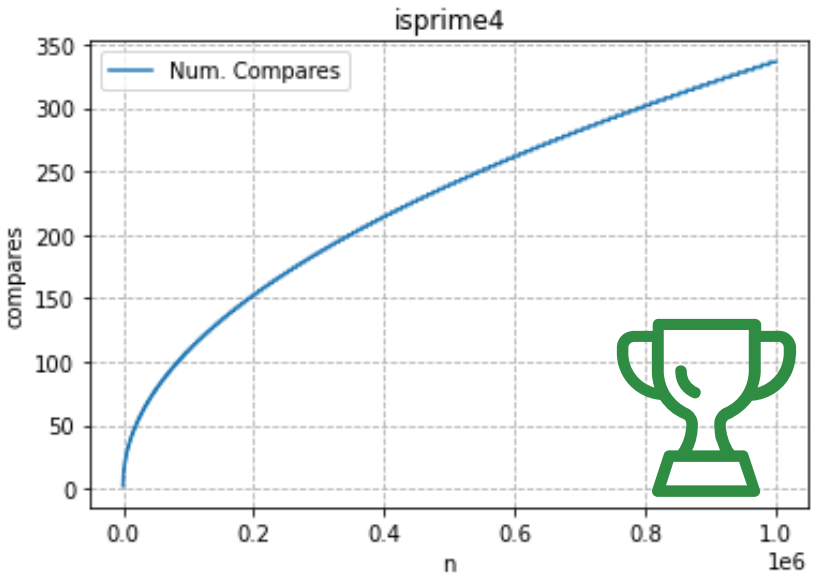
```java
public boolean isprime4(int n)
{
   if (n <= 1)
   {
      return false;
   }

   if (n <= 3)
   {
      return true;
   }

   if (n % 2 == 0 || n % 3 == 0)
   {
      return false;
   }

   for (int i = 5; i * i <= n; i += 6)
   {
      if (n % i == 0 || n % (i + 2) == 0)
      {
         return false;
      }
   }

   return true;
}
```



Can we do better?

isprime2 –VS– isprime4





Some Remarks

01

Knowledge

The more we know about our problems, the better we can design algorithms to solve them.

02

Logic Changes

A small change in a statement could make an algorithm to run better or worse.

02
Course Overview

How we run this course

Overall Course Topics

Algorithm Analysis

Thinking on resources required to run algorithms

Fundamental DS

Basic ideas for managing data

Sorting and Searching

Ordering and traversing data efficiently

Graphs

Connections between data points

Strings

Working with sequences of characters

Course Prerequisites

CS 180

Programming skills

...

CS 182

Analytical skills

...

CS 240, CS/STAT 242

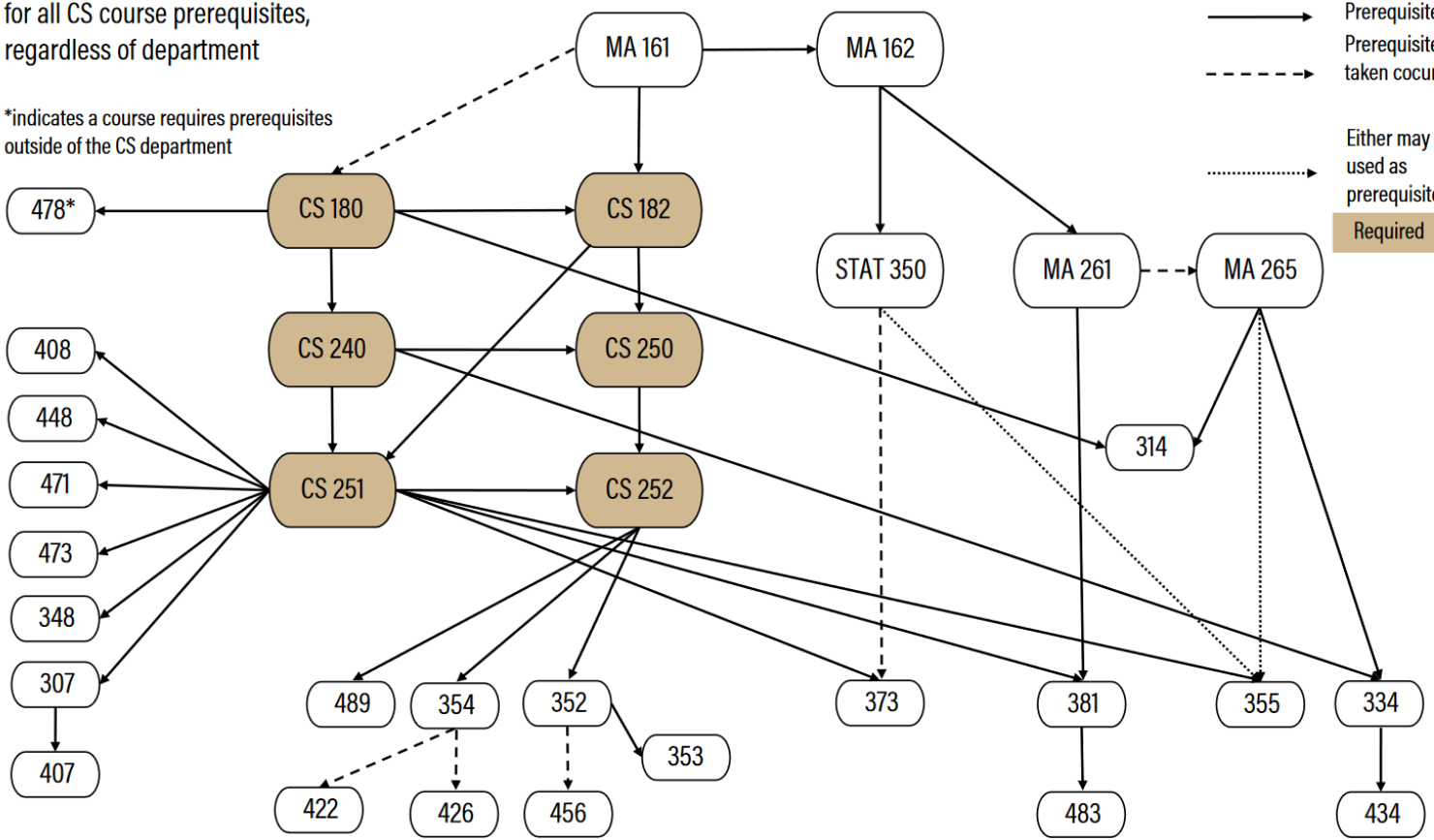
Coding Experience

...

A grade of 'C' or better is required for all CS course prerequisites, regardless of department

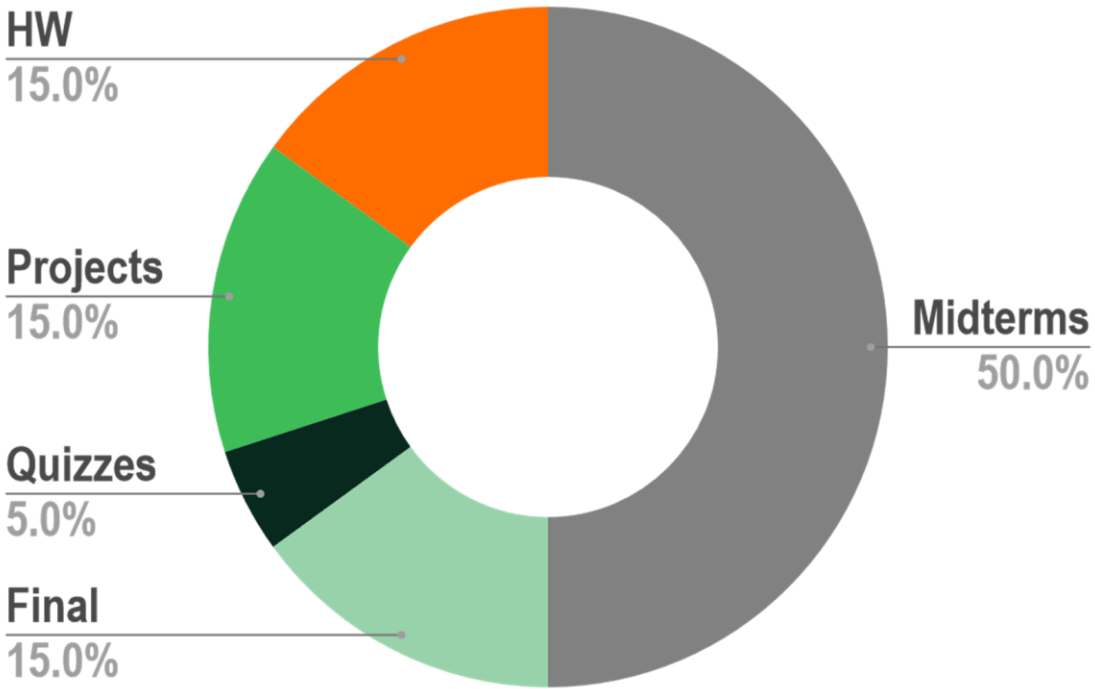
*indicates a course requires prerequisites outside of the CS department

Computer Science Prerequisite Flowchart



https://www.cs.purdue.edu/undergraduate/docs/CS-Prerequisite-Flowchart-202122.pdf

Grading



Specific information about each grading item in the syllabus.

Grading

Final Grades Distribution Guideline					
Grade	**Score**	**Grade**	**Score**	**Grade**	**Score**
A+	[99-100]	A	[93-99)	A-	[90-93)
B+	[86-90)	B	[83-86)	B-	[80-83)
C+	[76-80)	C	[73-76)	D or lower	[0-73)

We will determine the exact grading scale at the end of the semester.
All final grades will be assigned systematically.

Philosophy: Select cut-offs to reflect mastery of material/preparedness for courses building on 251 (No fixed % of A's, B's, C's,...)

Important Dates

1st Midterm

Friday, February 7th
Class time

2nd Midterm

Friday, March 7th
Class time

3rd Midterm

Friday, April 11th
Class time

MLK Jr. Day

Monday, January 20th
(Observed)

Spring Break

Monday - Friday
March 17th – 22nd

Final Exam

TBA
Finals Week

Course Platforms

Brightspace

Official course content and announcements platform

Ed Discussion

Online Q&A
Expect replies during working days/hours

Gradescope & Vocareum

For submitting homework assignments and programming projects

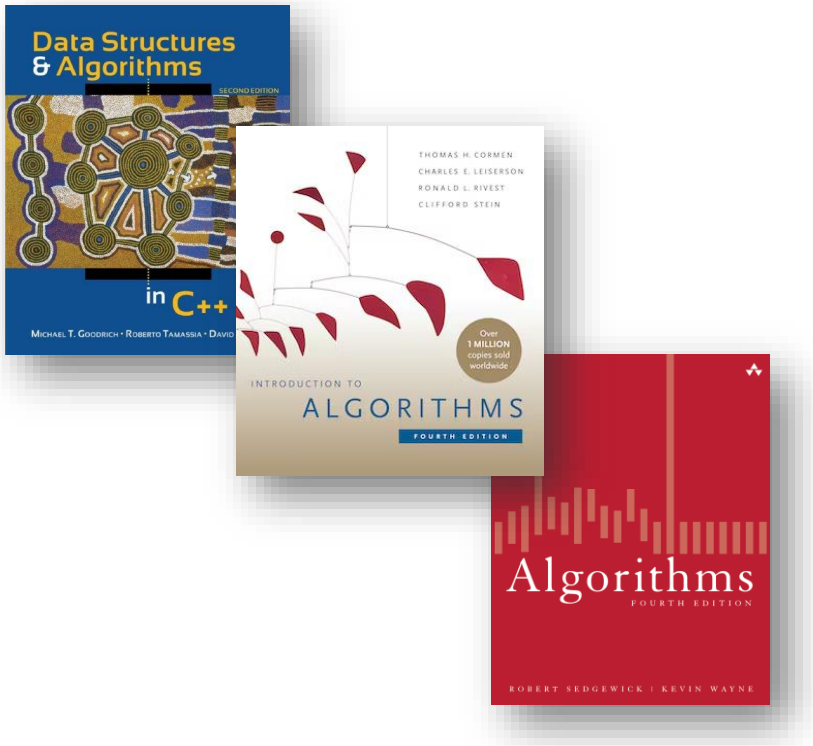
Seriously, do not send us emails with questions about topics or course activities. Use Ed instead (you can make private posts that only the course staff will see).

Recommended Textbooks

Data Structures & Algorithms in C++
Second Edition
By Goodrich, Tamassia, and Mount
John Wiley and Sons Inc.

Introduction to Algorithms
Fourth Edition
By Cormen, Leiserson, Rivest, and Stein
The MIT Press

Algorithms, Fourth Edition
By Robert Sedgewick and Kevin Wayne
Addison-Wesley



Recommended ≠ Required

How to Succeed in this Course?

Self-Discipline

Consider your time commitment as sacred.

Time Management

It is too late if you start the day before deadline.

Be Active!

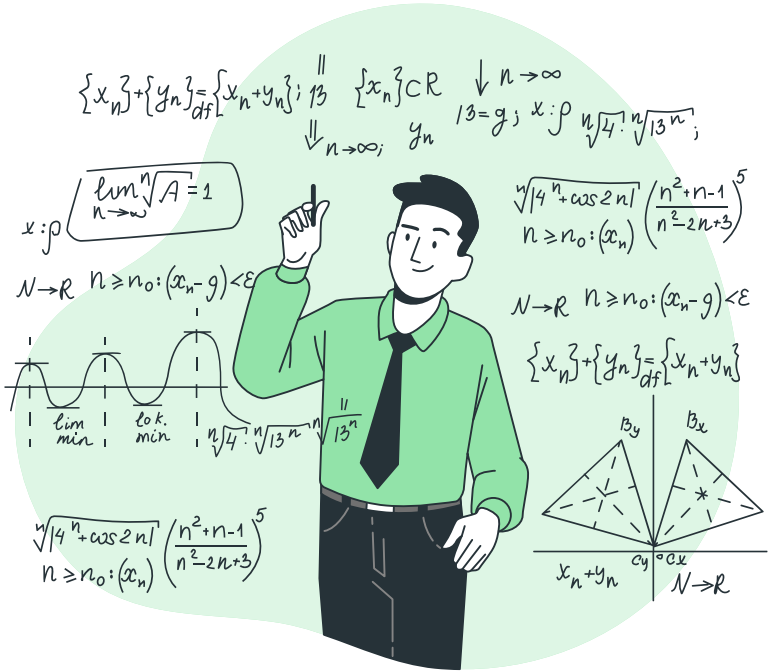
Learning DS & Algo. by watching? Good luck.

"Coding is to programming what typing is to writing. If you learn to program by learning to code, you essentially only know how to type."

— Leslie Lamport

https://www.youtube.com/watch?v=rkZzq7Vowao

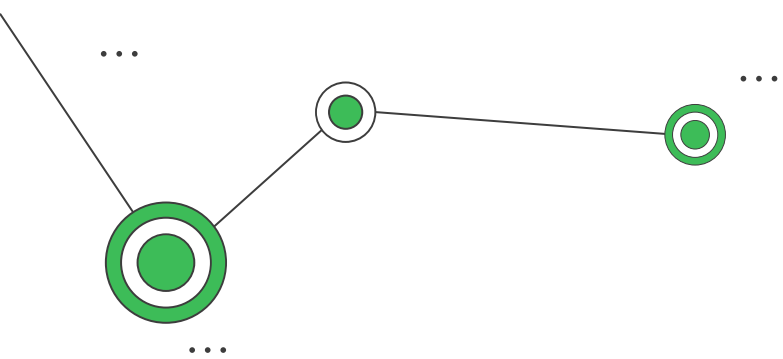
Invest in Math and Reasoning



Math and Reasoning dominate CS. Successful coding reflects good theoretical understanding.

Does something look to be true? Prove it!

- Direct/Indirect Proof
- Proof by Induction
- Proof by Construction
- Proof by Diagonalization



Example: $\sqrt{2}$ is Irrational

Assume $\sqrt{2}$ is rational. That is, it can be written as the ratio of $p, q \in \mathbb{Z}$ with no common factors (Irreducible):

$$\sqrt{2} = \frac{p}{q}$$

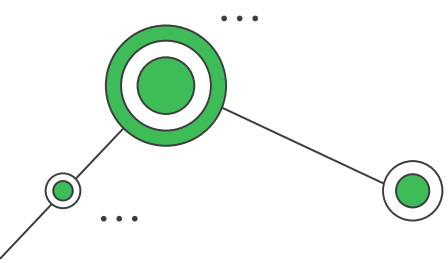
Square both sides: $2 = \frac{p^2}{q^2}$

Implies $p^2 = 2q^2$ (p^2 is even, which is true if p itself is even).

Since p is even, there exists a $k \in \mathbb{Z}$ such that $p = 2k$.

$4k^2 = p^2 = 2q^2$. Then $q^2 = 2k^2$.

Implies q^2 and q are even, which contradicts that $\frac{p}{q}$ is irreducible. By contradiction, $\sqrt{2}$ is irrational. **Q.E.D.**



Also, Code Everything!

If you understand the concepts, then you will know how to implement them.

"My code doesn't work, and I don't know why!" Then **debug it!**

PEBCAK: Problem Exists Between Chair and Keyboard.

Using AI in the Course

This is analogous to paying for a gym membership yet relying only on supplements to get results.

It is helpful for short-term results but awful for meaningful learning when misused.

"You don't want to be the 'professional' proposing range queries on a hash table."

Additional Information

Nondiscrimination Statement

We foster tolerance, sensitivity, understanding, and mutual respect. We proceed according to Purdue's policies in case of discriminatory actions.

Students with Disabilities or Diagnosed Conditions

We are eager to make your learning experience as meaningful as possible. Please contact the DRC and let us know how we can assist you.

Mental and Physical Health

Your health and safety are top priorities! Do not hesitate to ask for assistance when required. PUSH and ODOS have staff ready to help.

Additional Information

Academic Integrity

Do your work with honesty and integrity. **Ask for guidance instead of cheating**. We check for plagiarism and other forms of academic dishonesty.

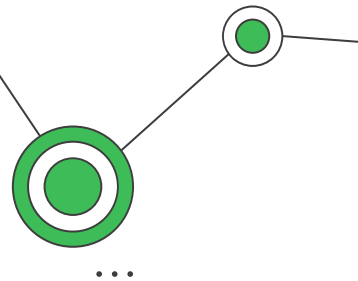
Technical Difficulties

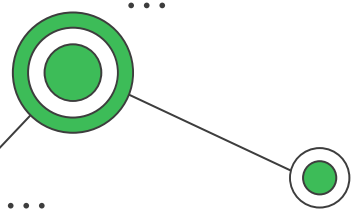
Cannot log in on Brightspace? Does Gradescope or Vocareum not work for you? Let us know ASAP (**no later than 15 minutes after the incident**).

Getting Assistance from TAs

TAs are eager to assist you. Be mindful as they are students just like you. Also, expect the TA rooms to be at full capacity as deadlines approach.

Read the Syllabus!
Quiz 1 is about it





CS251 – Data Structures and Algorithms
Spring 2025

The requirements and restrictions of this syllabus may change as we progress during the semester. Any official change will be announced via Brightspace only.

Course Information

CS25100 – Data Structures and Algorithms
Course credit hours: 3
Section: 13321 – LE1: Lilly Hall of Life Sciences 1105. MWF. 7:30 AM – 8:20 AM
From January 13, 2025, to May 3, 2025.

Instructor(s) and Course Staff Contact Information

Instructor(s):
Andres Bejarano – abejara@purdue.edu (read the Communication and Email Policy for more information)
Office Hours: Wednesday, 8:30 AM – 9:20 AM, DSAI 1119B (read the Office Hours Policy for more information)

Teaching Assistants: Check Brightspace for the list of TAs and their respective office hours information.

Course Description

This course offers a comprehensive introduction to key data structures and algorithms within computer science. Emphasis is placed on crafting efficient implementations, thoroughly understanding and contrasting various data structures and their integration into multiple algorithms. Students will learn to estimate the efficacy of these structures in real-world applications. The course aims to equip students with the skills to choose, create, and critically evaluate data structures for specific problems.

Some of the topics covered are Runtime analysis of algorithms, Primitive data structures, Heaps, Trees, Searching and Sorting, Binary Search Trees, Hashing, Graphs, Tries, and Spatial Data Structures

EOL – End of Lecture

Do you have any questions?