


Introduction to Computer Science: Programming Languages, Computation and Discrete Mathematics

James Aaron Erang  ¹

¹Biotechnology and Analytical Laboratory, Department of Biological Sciences, College of Science, Central Luzon State University

September 2024

Programming

Definition



Francesco Napoletano 

@napolux



Give a man a program, frustrate him for a day. Teach a man to program, frustrate him for a lifetime.

12:30 PM · 11 May 20 · [Buffer](#)

492 Retweets 1,759 Likes

Figure: Posted by u/Sour-Fun in r/ProgrammerHumor: Consequences and side effects of knowing ‘how’ to program a ‘*program*.’

Programming

Definition



Omar Hernandez · 3rd+

Software Developer
18h · 🌐

[+ Follow](#)

My mother had a look at my code today and said:

"So, your job is about writing text using random English words and symbols in fancy colors? I do not know why you get so well paid for that. That text is not even properly aligned to the left."

She would be a very good code reviewer.

👍👎❤️ 5,697

91 comments · 49 shares

What is Programming?

A method used to write a sequence of instructions and implementation of step by step specification of procedures designed to solve a class of specific problems or perform a computational task.

Figure: Posted by u/blumzzzz in r/ProgrammerHumor: What is “*programming*”?

Algorithms, Programs and Languages

Distinction

Algorithm, Program and Programming Languages

Algorithm(s) are step by step specifications of procedures or a finite sequence of mathematically rigorous instructions, designed to solve a class of specific problems or perform a computation. **Program**, on the other hand, is a sequence of instructions composed through **programming** using a system of notation utilized in coding or writing a computer program.

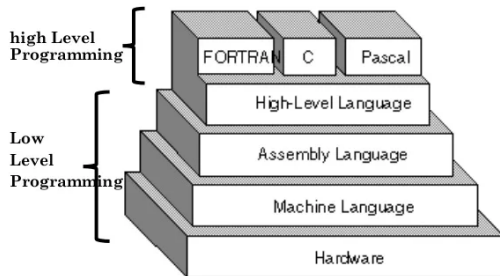


Figure: Classification of programming language based on its abstraction from computer details.

Code Equivalence

Pseudo-Code Implementation

Four programs were written to perform the following sequence of instructions:

- 1 Set the value "hello world!" to variable hello
- 2 Check if the value of variable hello is equal to "hello world!"
- 3 If the value is equal, print "The tradition in programming is to print" and print "Hello World! Like this in %s"
- 4 Then *loop* or *repeat* *i* times while printing "%i hello world!"
- 5 Finally, return a value of 0

All of the four programs are fundamentally reducible into similar function and underlines the same sequence of instructions:

```
FUNCTION main:
  SET VALUE "hello world!" to hello
  CHECK IF hello IS EQUAL TO "hello world!"
  IF EQUAL
    PRINT "The tradition in programming is to print"
    PRINT "Hello World! Like this in %s"

  LOOP i TIMES
    PRINT "%i hello world!"
  RETURN 0
```

Listing: Pseudo-code equivalence of programs exhibited above.

Code Equivalence

C and Rust Implementation

Programming language, as previous defined, is a system of notation that can be used to write algorithms or programs with specific function.

```
#include <stdio.h>
#include <string.h>

void main() { /* this is comment, this is ignored */
    char hello[] = "hello world!"; // this is variable

    if (strcmp(hello, "hello world!") == 0) {
        printf(
            "The tradition in programming is to print\n"
        );
        printf(
            "Hello World! Like this in C, %s\n", hello
        );

        for (int i = 1; i < 4; i++) { // loop over range of 1 to
            printf(                  // 4 and print hello world!
                "%d hello world!\n", i
            );
        }

        // return 0;
    }
}
```

Listing: Hello world! example in C.

```
fn main() {
    /* this is a comment */
    // this is ignored
    // declare a variable
    let hello = String::from("hello world!");

    // compare this variable to a string
    if hello == "hello world!" {
        println!(
            "The tradition in programming is to print"
        );
        println!(
            "Hello World! Like this in Python, {}", hello
        );

        // loop over range of 1 to 4 and print hello world!
        for i in 1..4 {
            println!(
                "{} hello world!", i
            );
        }
    }
}
```

Listing: Hello world! example in Rust.

Hence, similar functions or algorithms can be implemented in several programming languages.

Code Equivalence

C++ and Python Implementation

Consequently, due to **semantics** or **syntax** difference(s), various code formatting or style discrepancies are observable

```
#include <cstring>
#include <iostream>

int main() {
    /* this is long comment */
    // this is ignored
    // this is variable
    char hello[] = "hello world!";

    // compare this variable to a string
    if ( strcmp(hello, "hello world!") == 0 ) {
        std::cout << "The tradition in programming is to"
            << std::endl;
        std::cout << "print Hello World! Like this in C++"
            << hello
            << std::endl;

        for (int i = 0; i < 3; i++) {
            std::cout <<
                i << " " << hello
                << std::endl;
        }

        return 0;
    }
}
```

Listing: Hello world! example in C++.

```
class Main[Example]:
    def __init__(self) -> None:
        pass

    def main(self) -> int:
        """ this is a docstring """
        # this is ignored
        # this is a variable
        hello: str = "hello world!"

        # compare this variable to a string
        if hello == "hello world!":
            print(
                "The tradition in programming is to print"
            )
            print(
                f"Hello World! Like this in Python, {hello}"
            )

            for i in range(1, 4):
                print(f"{i} Hello world!")

        return 0

if __name__ == "__main__":
    Main().main()
```

Listing: Hello world! example in Python.

Algorithms in Detail

Binary Search

Given an array A of length n :

| | | | | | | | |
|-------|-------|-------|-------|-------|-------|---------|-----------|
| i_0 | i_1 | i_2 | i_3 | i_4 | i_5 | \dots | i_{n-1} |
|-------|-------|-------|-------|-------|-------|---------|-----------|

$$A = \begin{bmatrix} i_0 \\ i_1 \\ i_2 \\ i_3 \\ i_4 \\ i_5 \\ \dots \\ i_{n-1} \end{bmatrix}$$

How can the index i_n of the curative compound C be determined?

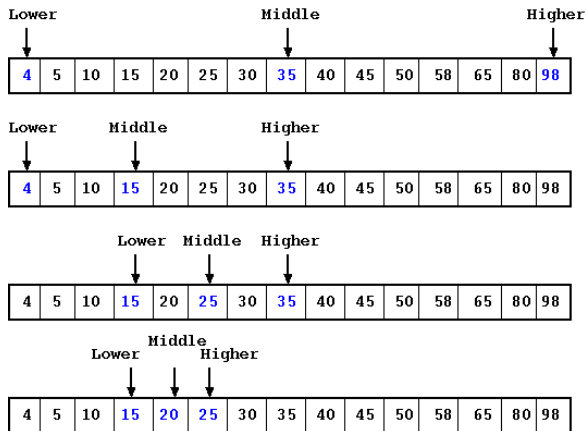


Figure: Mechanism of binary search algorithm in array of length N .

Algorithms in Detail

Binary Search

```
class BinarySearch:
    """ Binary search solution in Python """

    def __init__(self, arr: list[int]) -> None:
        self.arr: list[int] = arr

    def solution(self, T: int) -> int:
        """ Solution """

        while (L := 0) <= (H := (len(arr) - 1)):
            M: int = (H + L) // 2

            if self.arr[M] < T:
                L = M + 1
            elif self.arr[M] > T:
                H = M - 1
            else: # if self.arr[M] == T
                return M

        return 1
```

Listing: Implementation of binary search in Python
3.12

The algorithm involves several specified steps:

- 1 Set $L = 0$ and $H = n - 1$
- 2 Find the middle index M :

$$M = \left\lfloor \frac{(L + H)}{2} \right\rfloor$$

- 3 If $A_M < T$; set $L = M + 1$
- 4 However, if $A_M > T$; set $H = M - 1$
- 5 Return M if $A_M = T$
- 6 If $L > n$ return 1

Algorithms in Detail

Index of MAGAININ 1 in ChEMBL Query

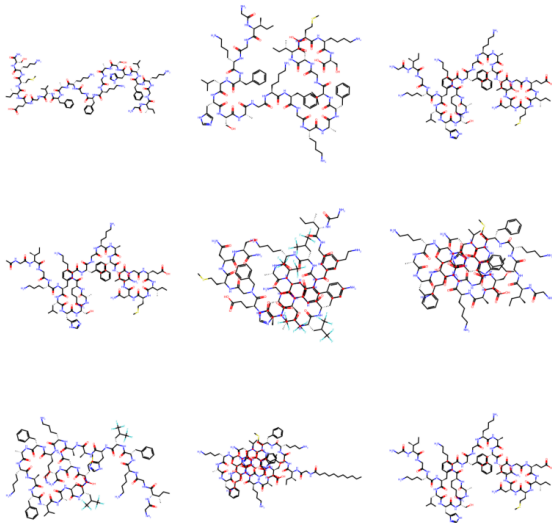


Figure: Molecular structures derived from SMILES (Simplified Molecular Input Line Entry System) of molecules retrieved from ChEMBL database using “MAGAININ” as query.

Algorithms in Detail

Index of MAGAININ 1 in ChEMBL Query

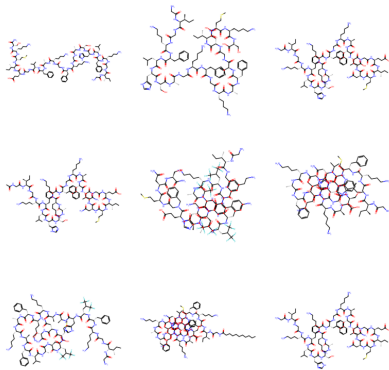


Figure: Molecular structures derived from SMILES of molecules retrieved from ChEMBL database using "MAGAININ" as query.

Associated two 1×9 arrays with the response of database.

$$A = \begin{bmatrix} \text{ChEMBL437357} \\ \text{ChEMBL409372} \\ \text{ChEMBL414933} \\ \text{ChEMBL4088094} \\ \text{ChEMBL1673385} \\ \text{ChEMBL1673394} \\ \text{ChEMBL1673389} \\ \text{ChEMBL1673395} \\ \text{ChEMBL412693} \end{bmatrix} \quad B = \begin{bmatrix} \text{Protein} \\ \text{Protein} \\ \text{Protein} \\ \text{Small Molecule} \\ \text{Protein} \\ \text{Protein} \\ \text{Protein} \\ \text{Small Molecule} \\ \text{Protein} \end{bmatrix}$$

Mathematics in Computer Science

Expression of Mathematical Equations in Programming

The saturation function r is defined as the quotient from the portion of bound ligand to the total amount of the macromolecule $r = [L]_{\text{bound}}/[M]_0$:

$$r = \frac{\sum_{i=1}^n i \binom{n}{i} \left(\frac{[L]}{K_D}\right)^i}{1 + \sum_{i=1}^n \binom{n}{i} \left(\frac{[L]}{K_D}\right)^i}$$
$$\binom{n}{i} = \frac{n!}{(n-i)!i!}$$

Equation Set 1: Saturation function r of bound ligand to the macromolecule .

```
#include <math.h>

unsigned int factorial(int N) {
    if (N <= 1) { return 1; }
    else {
        int prod = 1;
        for (int k = 1; k <= N; k++) {
            prod *= k;
        }
        return prod;
    }
}

int r_sat_func() {
    float r, r_num, r_den;
    // example arbitrary value, does not represent any
    float L = 50.05; float Kd = 1.65;
    int n = 5; int i = 4;

    unsigned int n_i_matrix =
        factorial(n)/factorial(n - i) * factorial(i);

    for (int j = 1; j == n; j++) {
        r_num += i * n_i_matrix * pow((L/Kd), i);
        r_den += 1 + n_i_matrix * pow((L/Kd), i);
    }
    return r_num + r_den;
}
```

Listing: Implementation of saturation function (Equation Set 1) in C.