Basics of Competitive programming

By: Riham Katout

Introduction

- What is competitive programming?

Competitive programming is a sport where contestants solve algorithmic problems within a time limit using a programming language of their choice. It tests problem-solving skills, knowledge of algorithms, and ability to write efficient code.

It’s benefits:

1. Problem-Solving and Programming Skills
2. Technical Interview Preparation
3. Prizes
4. Enhances Your Thinking Ability
5. Unleashes the Ability to Write Better and Optimized Code

Competitive programming problems

1. Searching
2. Sorting
3. Dynamic Programming
4. Graph Theory
5. Number Theory
6. Greedy Algorithm
7. Backtracking
8. Divide and Conquer
9. String Manipulation
10. Data Structure

Set up your environment

1. Choose a suitable language, C++ is recommended
2. Install a good IDE (visual studio, visual code, code blocks are good)
3. Install required libraries and extensions
4. Enjoy your cup of tea ☺

How to install IDE

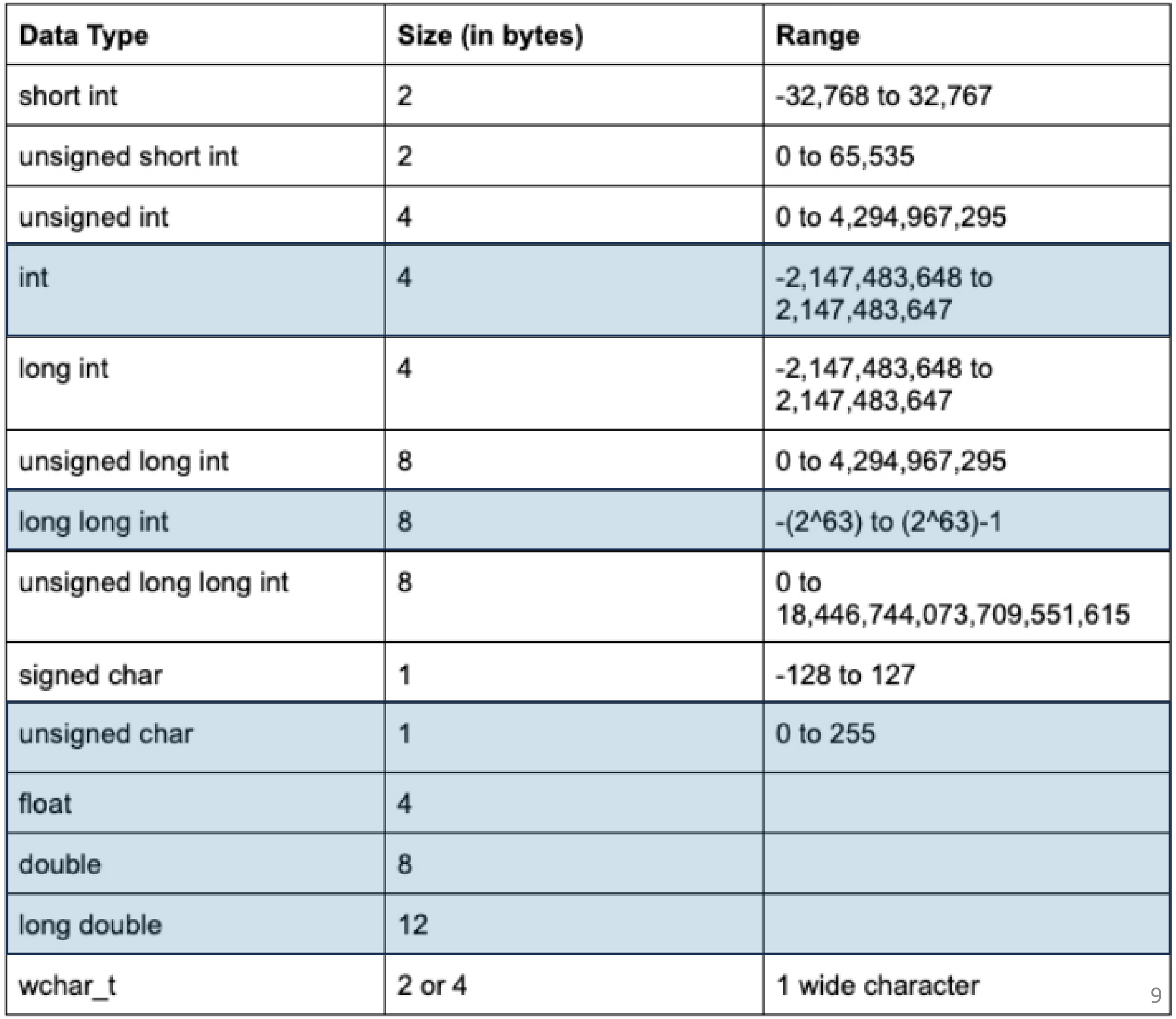
- visual studio for C++
- visual studio code
- codeblocks

Basics of C++

Start by watching these videos

1. Introduction
2. First project in C++
3. Escape sequence
4. variables vs data type

Build in C++ Data Types

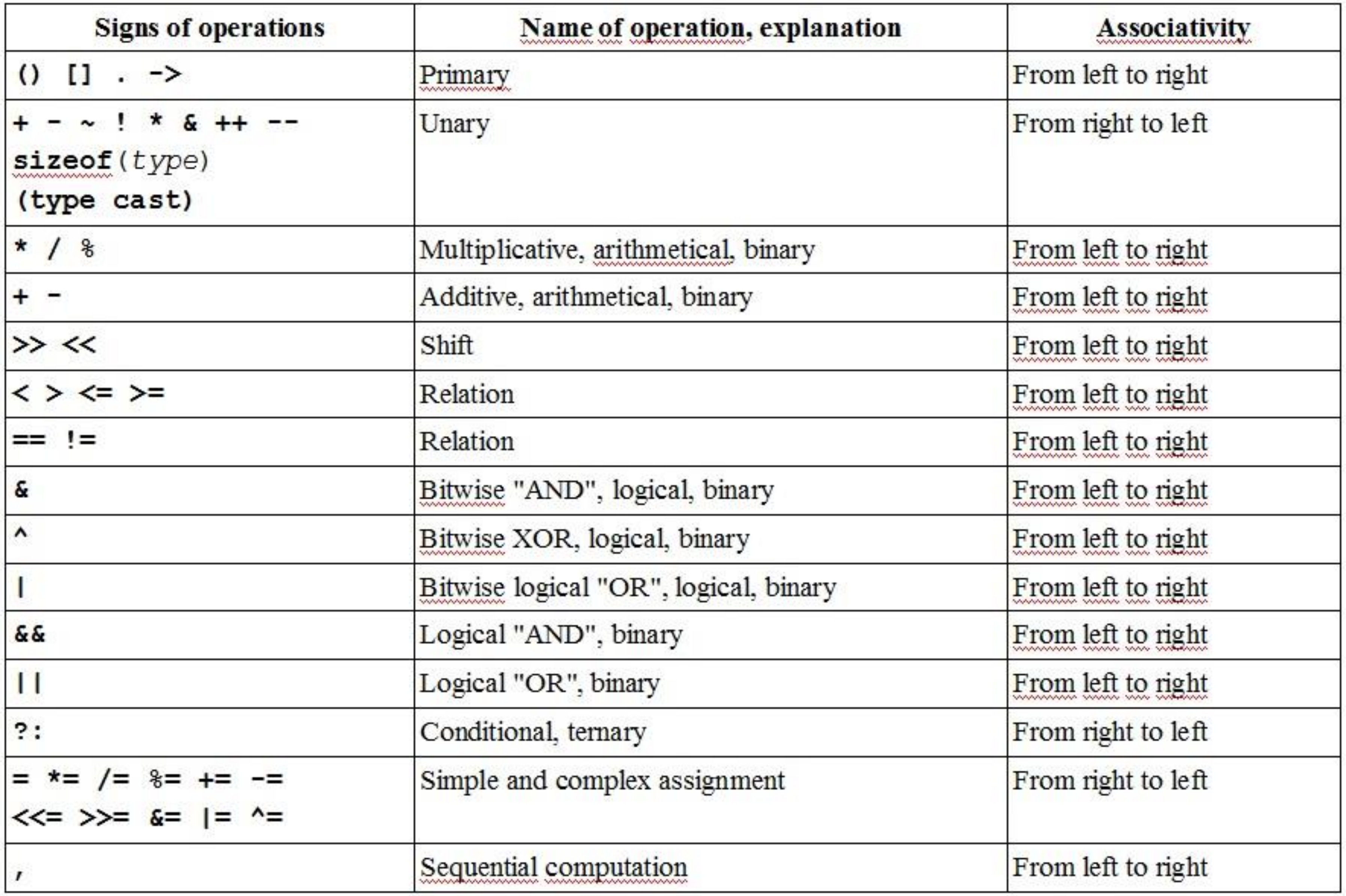


Data Type	Size (in bytes)	Range
short int	2	-32,768 to 32,767
unsigned short int	2	0 to 65,535
unsigned int	4	0 to 4,294,967,295
int	4	-2,147,483,648 to 2,147,483,647
long int	4	-2,147,483,648 to 2,147,483,647
unsigned long int	8	0 to 4,294,967,295
long long int	8	-(2^63) to (2^63)-1
unsigned long long int	8	0 to 18,446,744,073,709,551,615
signed char	1	-128 to 127
unsigned char	1	0 to 255
float	4	
double	8	
long double	12	
wchar_t	2 or 4	1 wide character

Videos to watch

1. Priorities and calculations
2. Basic Arithmetic & casting
3. Prefix & postfix compound
4. Variable scope (local vs. global)

Priorities & Calculations



Signs of operations	Name of operation, explanation	Associativity
`() [] . ->`	Primary	From left to right

`+ - ~ ! * & ++ --`
`sizeof(type)`
`(type cast)`
	Unary	From right to left

`* / %`	Multiplicative, arithmetical, binary	From left to right
`+ -`	Additive, arithmetical, binary	From left to right
`>> <<`	Shift	From left to right
`< > <= >=`	Relation	From left to right
`== !=`	Relation	From left to right
`&`	Bitwise "AND", logical, binary	From left to right
`^`	Bitwise XOR, logical, binary	From left to right
`|`	Bitwise logical "OR", logical, binary	From left to right
`&&`	Logical "AND", binary	From left to right
`||`	Logical "OR", binary	From left to right
`?:`	Conditional, ternary	From right to left

`= *= /= %= += -=`
`<<= >>= &= |= ^=`
	Simple and complex assignment	From right to left

`,`	Sequential computation	From left to right

Solve problems is the best way to practice ☺

It may be hard at the beginning, it’s OK you won’t die ^_^

Remember it’s a new thing so the normal situation is being difficult but you can ask for help in our communities

[Palestinian community]

[Najah National University Community]

We'll solve problems on the following sites

- Hackerrank → how to register and use it (at 2:40 min)
- Codeforces → how to register and use it
- Atcoder → how to register and use it

Basic problems

1. Hackerrank - say Hello, World!
2. Hackerrank - print sum
3. Hackerrank - Data types example
4. Codeforces - domino piling
5. Codeforces - drinks

Selection statements

if statement, switch statement

Selection statements videos

1. if statement
2. logical operators
3. switch statement

Selection statements problems

1. Hackerrank - Conditional statements
2. Codeforces – watermelon
3. Codeforces - Soldier and Bananas
4. Codeforces - Lucky division
5. Codeforces - Theatre square
6. Codeforces - Cheap travel
7. Codeforces - Even odds

Loops

for, while, do-while

Loops videos

1. While, do-while loops
2. For loop
3. Examples of loop, break, continue
4. Nested loop
5. Draw shapes (triangle)
6. Draw shapes (square & some letters)

Loops problems

1. Hackerrank - For Loop
2. Codeforces - A+B
3. Codeforces - Young physicist
4. Codeforces – Hulk
5. Codeforces - Vanya and fence
6. Codeforces - Bear and big brother
7. Codeforces - Wrong subtraction
8. Codeforces - Kefa and first steps
9. Codeforces – Taxi **important
10. Codeforces - Good array
11. Atcoder - Find Takahashi
12. Codeforces - The day of Pi

Functions

Functions videos

1. Part 1
2. Part 2
3. Built in functions
4. Random function
5. Call by reference vs. call by value
6. Recursion 1
7. Recursion 2
8. Default arguments

Frequently used built in functions

#include <math> functions

Function	Description	Example

`ceil( x )`	rounds *x* to the smallest integer not less than *x*	`ceil( 9.2 )` is `10.0`
		`ceil( -9.8 )` is `-9.0`

`cos( x )`	trigonometric cosine of *x* (*x* in radians)	`cos( 0.0 )` is `1.0`

`exp( x )`	exponential function e^x	`exp( 1.0 )` is `2.71828`
		`exp( 2.0 )` is `7.38906`

`fabs( x )`	absolute value of *x*	`fabs( 5.1 )` is `5.1`
		`fabs( 0.0 )` is `0.0`
		`fabs( -8.76 )` is `8.76`

`floor( x )`	rounds *x* to the largest integer not greater than *x*	`floor( 9.2 )` is `9.0`
		`floor( -9.8 )` is `-10.0`

`fmod( x, y )`	remainder of *x*/*y* as a floating-point number	`fmod( 2.6, 1.2 )` is `0.2`

`log( x )`	natural logarithm of *x* (base *e*)	`log( 2.718282 )` is `1.0`
		`log( 7.389056 )` is `2.0`

`log10( x )`	logarithm of *x* (base 10)	`log10( 10.0 )` is `1.0`
		`log10( 100.0 )` is `2.0`

`pow( x, y )`	*x* raised to power *y* (x^y)	`pow( 2, 7 )` is `128`
		`pow( 9, .5 )` is `3`

`sin( x )`	trigonometric sine of *x* (*x* in radians)	`sin( 0.0 )` is `0`
`sqrt( x )`	square root of *x* (where *x* is a nonnegative value)	`sqrt( 9.0 )` is `3.0`
`tan( x )`	trigonometric tangent of *x* (*x* in radians)	`tan( 0.0 )` is `0`

Functions problems

1. Atcoder – Power
2. Codeforces - Pens and pencils **ceil function
3. Codeforces - Extremely round **try to solve it using log10 function
4. Codeforces - Dreamoon and stairs
5. Codeforces – Factorial **try to solve it using recursion
6. Atcoder - A recursive function
7. Codeforces - Stand-up Comedian ** try to use min, max functions in <algorithm> library
8. Codeforces - Cardboard for Pictures ** sqrt function

Arrays

1D & 2D

Arrays videos

1. 1D array - part 1
2. 1D array - part 2
3. 1D array - part 3 - passing array to function
4. 1D array - part 4 - array of charachters
5. 2D array

Array problems

1. Hackerrank - Arrays introduction
2. Hackerrank - Variable sized array
3. Codeforces – Puzzles **search for sort function in <algorithm>
4. Atcoder - Shift
5. Atcoder - Sequence of strings **try to use reverse function in <algorithm>
6. Codeforces - Beautiful matrix

Pointers

Pointers videos

1. Part 1
2. Part 2

Structure

Structure video

1. Data Structure - Struct

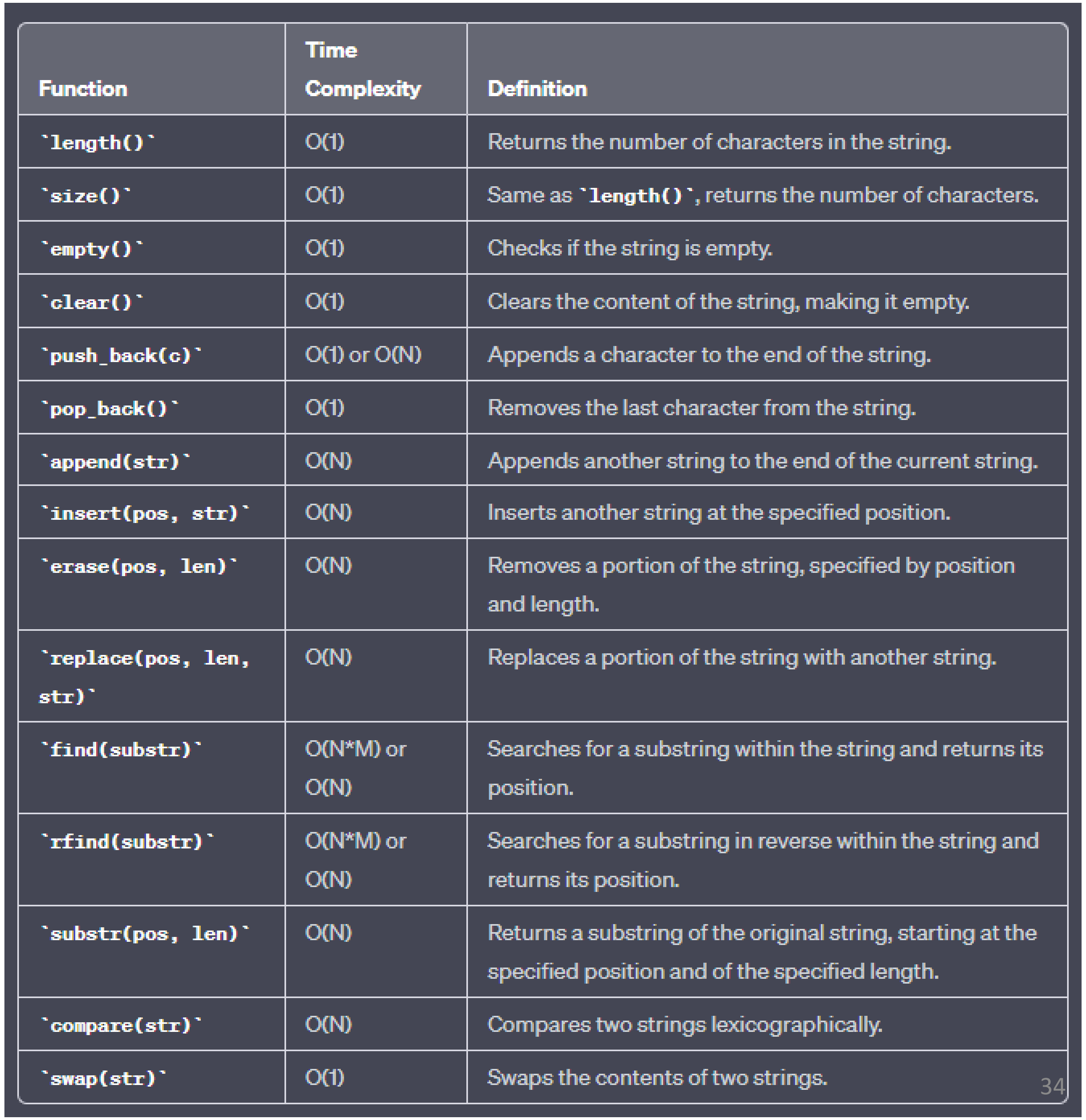
String

String video

1. String introduction

String Functions

Here is the most commonly used functions in <string> library.



Function	Time Complexity	Definition
`length()`	O(1)	Returns the number of characters in the string.
`size()`	O(1)	Same as `length()`, returns the number of characters.
`empty()`	O(1)	Checks if the string is empty.
`clear()`	O(1)	Clears the content of the string, making it empty.
`push_back(c)`	O(1) or O(N)	Appends a character to the end of the string.
`pop_back()`	O(1)	Removes the last character from the string.
`append(str)`	O(N)	Appends another string to the end of the current string.
`insert(pos, str)`	O(N)	Inserts another string at the specified position.
`erase(pos, len)`	O(N)	Removes a portion of the string, specified by position and length.
`replace(pos, len, str)`	O(N)	Replaces a portion of the string with another string.
`find(substr)`	O(N*M) or O(N)	Searches for a substring within the string and returns its position.
`rfind(substr)`	O(N*M) or O(N)	Searches for a substring in reverse within the string and returns its position.
`substr(pos, len)`	O(N)	Returns a substring of the original string, starting at the specified position and of the specified length.
`compare(str)`	O(N)	Compares two strings lexicographically.
`swap(str)`	O(1)	Swaps the contents of two strings.

String problems

1. Atcoder - wwwvvvvvv
2. Codeforces - Anton and Polyhedrons
3. Codeforces - Way Too Long Words
4. Codeforces - Queue at the School
5. Codeforces - Dubstep
6. Codeforces - String Task **tolower function
7. Codeforces - Boy or Girl **frequency array
8. Codeforces – Football
9. Codeforces - Chat room
10. Codeforces - Translation
11. Codeforces - Amusing Joke
12. Codeforces - Anton and Letters