Big-O notation

Competitive Programming: Core Skills

Artur Riazanov

SPbSU

Introduction

- In this video we will master powerful technique for measuring the number of atomic operations an algorithm does.

Introduction

- In this video we will master powerful technique for measuring the number of atomic operations an algorithm does.
- The problem is that the number of operations depends on input.

Introduction

- In this video we will master powerful technique for measuring the number of atomic operations an algorithm does.
- The problem is that the number of operations depends on input.
- We will learn how to measure the dependence.

What is atomic operation?

When we estimate the number of operation it's useful to define what operation is.

What is atomic operation?

When we estimate the number of operation it's useful to define what operation is.

- Arithmetic operations (+, -, *, /, %, <, >, =)

What is atomic operation?

When we estimate the number of operation it's useful to define what operation is.

- Arithmetic operations (+, -, *, /, %, <, >, =)
- Logical operations (`or`, `and`, `not`, `xor`)

What is atomic operation?

When we estimate the number of operation it's useful to define what operation is.

- Arithmetic operations (+, -, *, /, %, <, >, =)
- Logical operations (`or`, `and`, `not`, `xor`)
- Accessing a value from an array

What is atomic operation?

When we estimate the number of operation it's useful to define what operation is.

- Arithmetic operations (+, -, *, /, %, <, >, =)
- Logical operations (`or`, `and`, `not`, `xor`)
- Accessing a value from an array
- Defining a new variable

What is atomic operation?

When we estimate the number of operation it's useful to define what operation is.

- Arithmetic operations (+, -, *, /, %, <, >, =)
- Logical operations (`or`, `and`, `not`, `xor`)
- Accessing a value from an array
- Defining a new variable

Which operations are not atomic?

Some operations seem simple but in fact they are pretty complicated and require much more time that atomic operations do.

- Comparing strings, vectors (C++) or lists (Python)

Which operations are not atomic?

Some operations seem simple but in fact they are pretty complicated and require much more time that atomic operations do.

- Comparing strings, vectors (C++) or lists (Python)
- Defining a vector or list with many elements

Which operations are not atomic?

Some operations seem simple but in fact they are pretty complicated and require much more time that atomic operations do.

- Comparing strings, vectors (C++) or lists (Python)
- Defining a vector or list with many elements
- Concatenating two strings

Which operations are not atomic?

Some operations seem simple but in fact they are pretty complicated and require much more time that atomic operations do.

- Comparing strings, vectors (C++) or lists (Python)
- Defining a vector or list with many elements
- Concatenating two strings

Strings, vectors and lists consists of small elements therefore the operations above are applied for each elements of a big object (each symbol of a string, each element of a vector/list).

Substring

$n = \text{length}(s);\ m = \text{length}(t)$

```
for i in range(m-n+1):   (0, 1, ..., m − n)
    match = True
    for j in range(n):
        if s[j] != t[i+j]:   contradiction!
            match = False
            break   no need to check latter symbols
    if match:
        break
```