

Competitive Programming Final Project {

[Brute Force Algorithm]

// By: Emma Chetan and Mikaela Tereshkov

}



Table Of 'Contents' {

01 Intro to Brute Force

02 Dreamer

03 Ant Typing

}



Intro to Brute Force

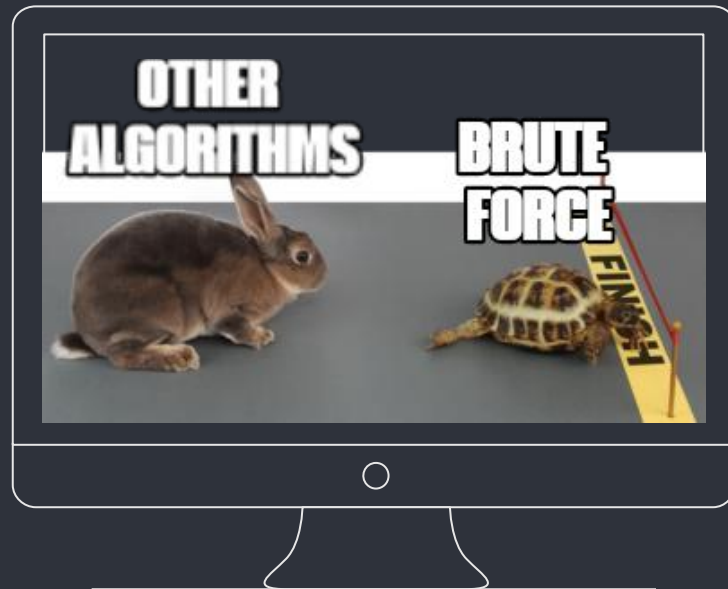
What is Brute Force?

Brute force is an algorithm for solving problems involving trying all reasonable options.

It involves checking many different options to see which would work.

Brute force can sometimes also include eliminating answers that are fated to be wrong if the problem permits.

Brute force relies on the speed of the computer, as it is typically a slower algorithm compared to other algorithms that can be used to solve some problems, though in some scenarios brute force is the only option.



Intro to Brute Force

Pros and Cons of Brute Force {

Pros {

- Exhaustive
- Simple, **easy-to-learn** idea
- Always provides a **correct** solution
- Versatile

Cons {

- Time complexity
- Redundancy; can be **inefficient**
- Requires more **memory** usage
- Not as **scalable** as some other **algorithms**

}

}



Intro to Brute Force

Outline of 'Brute Force' {

Step 01 Clearly understand the problem and all restraints

Step 02 Generate and iterate through all possible solutions

Step 03 Test whether it meets all required conditions

Step 04 Select the optimum, correct solution

Step 05 Analyze the time complexity

}



1
2
3
4
5
6
7
8
9
10
11
12
13
14



Dreamer {

Find the closest date on or after January 1st, 2000, Bash's birthday. We receive an **input**, but the digits are jumbled. We need to **output** the number of possible dates after or on his birthday and a rearrangement of these digits to show the closest possible date to 01 01 2000.

Ex **Input**: 01 01 0002

Ex **Output**: 4 01 01 ↓000

Key idea: Try out all possible **permutations** (**next_permutation**), use conditionals to isolate the possible valid dates (**backtracking**).

- a) **Brute force** to find all valid dates
- b) Year takes precedence over months, months over days
- c) **lower_bound** function on Bash's birthday finds the result

}



Ant Typing {

Calculate minimum number of seconds to type out a string of numbers on a keyboard with any order of the numbers starting on the leftmost key. Any action takes 1 second.

Catch: time limit of 1 second and maximum string length of 100000.

Key observation: though user cannot pre-calculate the result, they can group together pairs of two keys to pre-process part of the solution, checking each pair rather than the whole string, lowering iteration amount.

90 iterations per permutation (adding up to a worst case of 32659200 iterations), rather than 100000 (adding up to a worst case of 36288000000).

}



brute.cpp

force.cpp

1
2
3
4
5
6
7
8
9
10
11
12
13
14

