

Homework

111 Fall - Introduction to Algorithms

Outlines

Introduction to Algorithms / HW1

- Implementation Procedure
 - Account Registration & Login
 - Problem Solving
 - Screenshot of Verdict Record
- Uploading Rules
- Scoring Criteria
- Intro and Important Dates
- Homework Problems

Implementation Procedure

Online Judge Account Registration & Login (1/2)

- Register an Online Judge account with the link
(https://onlinejudge.org/index.php?option=com_comprofiler&task=registers).

The screenshot displays the Online Judge registration page. The browser's address bar shows the URL: onlinejudge.org/index.php?option=com_comprofiler&task=registers. The page layout includes a top navigation bar with various links, a left sidebar with a 'Login' section and a 'Main Menu', and a right sidebar with 'Our Patreons', 'Diamond Sponsors', 'Gold Sponsors', 'Silver Sponsors', 'Bronze Sponsors', and a 'Contribute' section.

The central registration form contains the following fields and options:

- Registration Form:**
 - Name:
 - Email:
 - Username:
 - Password:
 - Verify Password:
 - Former UVA ID:
 - Results email: ☐
 - Virtual Judge: ☐
 - Checkbox: ☐ 我不是机器人 (I am not a robot)
 - reCAPTCHA:
 - Your IP address is: 101.136.187.106
 - Register:

A red arrow points to the 'Register' button.

Implementation Procedure

Online Judge Account Registration & Login (2/2)

- Login with the link (<https://onlinejudge.org/>).

The screenshot shows the Online Judge website interface. At the top, there is a navigation bar with various icons and links. Below this, the main content area is divided into several sections:

- Login:** A form with fields for Username (ericitw) and Password (*****). There is a checkbox for "Remember me" and a "Login" button. Below the login button are links for "Forgot login?" and "No account yet? Register". A red arrow points to the "Login" button.
- Books and more books!:** A section featuring a book cover titled "ALGORITMOS ILUMINADOS" by Tim Roughgarden. The text mentions a new Spanish version of Part 1 of the "Algorithms Illuminated" series.
- Main Menu:** A sidebar menu with links to Home, Contact Us, and ICPC Live Archive.
- Online Judge:** A section with links to My uHunt with Virtual Contest Service, Browse Problems, Quick access, info and search, Problemsetters' Credits, Live Rankings, Site Statistics, Contests, Electronic Board, Additional Information, and Other Links.
- Our Patreons:** A section listing sponsors, including Diamond Sponsors (Steven & Felix Halim, Reinardus Praditya), Gold Sponsors (--- YOUR NAME HERE ---), Silver Sponsors (--- YOUR NAME HERE ---), and Bronze Sponsors (Christianto Handjo, Krzysztof Adamek, Fatima Broom, Amal Augustine).
- Contribute:** A section with a "Become a patron" button and a "Donate" button.

Implementation Procedure

Problem Solving (1/2)

- Go to the problem page (we will give you the link for each problem in the last section, for example, https://onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&page=show_problem&problem=36) to read the problem description.

The screenshot shows the Online Judge website interface. The browser address bar displays the URL: `onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&page=show_problem&problem=36`. The page title is "100 - The $3n + 1$ problem". The problem description text is as follows:

Problems in Computer Science are often classified as belonging to a certain class of problems (e.g., NP, Unsolvable, Recursive). In this problem you will be analyzing a property of an algorithm whose classification is not known for all possible inputs.

Consider the following algorithm:

1. input n
2. print n
3. if $n = 1$ then STOP
4. if n is odd then $n \leftarrow 3n + 1$
5. else $n \leftarrow n/2$
6. GOTO 2

Given the input 22, the following sequence of numbers will be printed

22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

It is conjectured that the algorithm above will terminate (when a 1 is printed) for any integral input value. Despite the simplicity of the algorithm, it is unknown whether this conjecture is true. It has been verified, however, for all integers n such that $0 < n < 1,000,000$ (and, in fact, for many more numbers than this).

Given an input n , it is possible to determine the number of numbers printed before and including the 1 is printed. For a given n this is called the *cycle-length* of n . In the example above, the cycle length of 22 is 16.

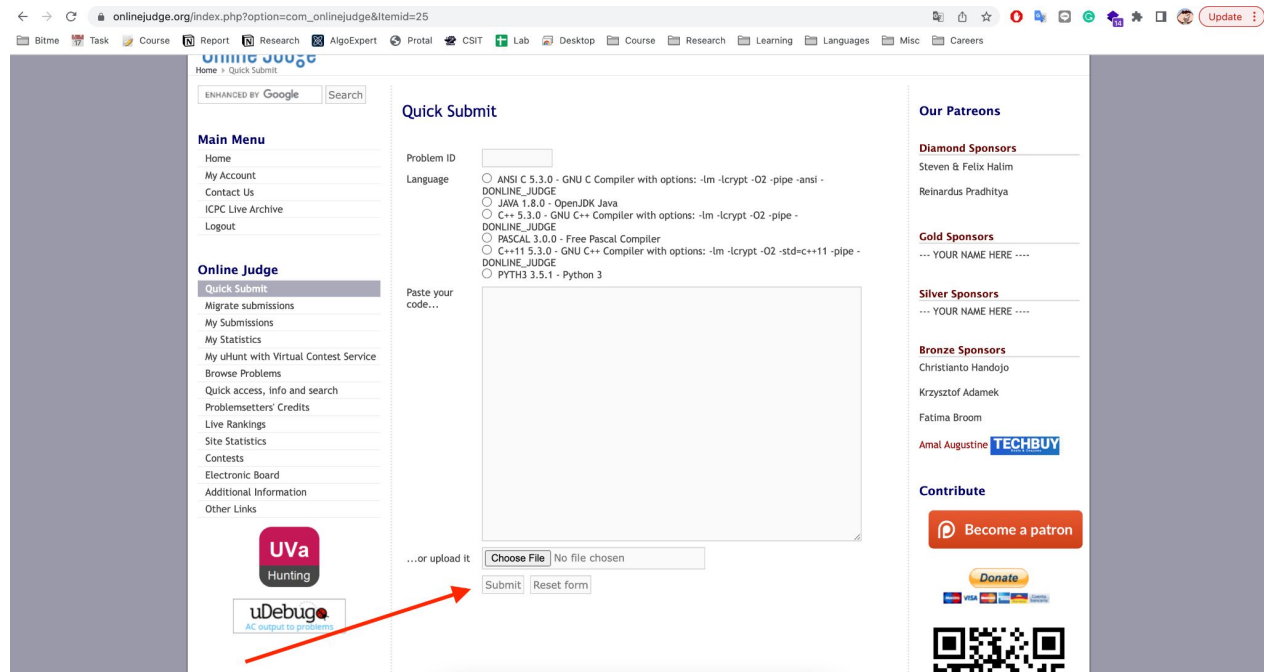
For any two numbers i and j you are to determine the maximum cycle length over all numbers between and including both i and j .

The interface also includes a left sidebar with navigation links, a top navigation bar, and a right sidebar with sponsor information.

Implementation Procedure

Problem Solving (2/2)

- After finish coding locally, submit code on this page (https://onlinejudge.org/index.php?option=com_onlinejudge&Itemid=25).



The screenshot shows the 'Quick Submit' page on the Online Judge website. The browser address bar displays 'onlinejudge.org/index.php?option=com_onlinejudge&Itemid=25'. The page layout includes a top navigation bar with links like 'Home', 'Quick Submit', 'Main Menu', and 'Online Judge'. The 'Main Menu' section on the left lists various links such as 'Home', 'My Account', 'Contact Us', 'ICPC Live Archive', and 'Logout'. The 'Online Judge' section lists links like 'Quick Submit', 'Migrate submissions', 'My Submissions', 'My Statistics', 'My uHunt with Virtual Contest Service', 'Browse Problems', 'Quick access, info and search', 'Problemsetters' Credits', 'Live Rankings', 'Site Statistics', 'Contests', 'Electronic Board', 'Additional Information', and 'Other Links'. The 'Quick Submit' form in the center has a 'Problem ID' field, a 'Language' dropdown menu with options for C, Java, C++, Pascal, and Python, and a large text area for pasting code. Below the code area are buttons for 'Choose File', 'Submit', and 'Reset form'. A red arrow points from the 'uDebug' logo at the bottom left to the 'Submit' button. The right sidebar contains sections for 'Our Patreons', 'Diamond Sponsors', 'Gold Sponsors', 'Silver Sponsors', 'Bronze Sponsors', and a 'Contribute' section with a 'Become a patron' button and a QR code.

Implementation Procedure

Screenshot of Verdict Record

- Go to submission page
(https://onlinejudge.org/index.php?option=com_onlinejudge&Itemid=9), screenshot the accepted record area for specific problem like [bottom image](#).

The screenshot shows the 'My Submissions' page on Online Judge. The table below is the data shown in the screenshot:

#	Problem	Verdict	Language	Run Time	Submission Date
27585182	100 The 3n + 1 problem	Wrong answer	C++	0.000	2022-06-20 14:29:53
27585171	100 The 3n + 1 problem	Accepted	C++	0.280	2022-06-20 14:27:48

The bottom row is highlighted with a red box, and a red arrow points from it to the bottom of the slide.

27585171 100 The 3n + 1 problem


Accepted

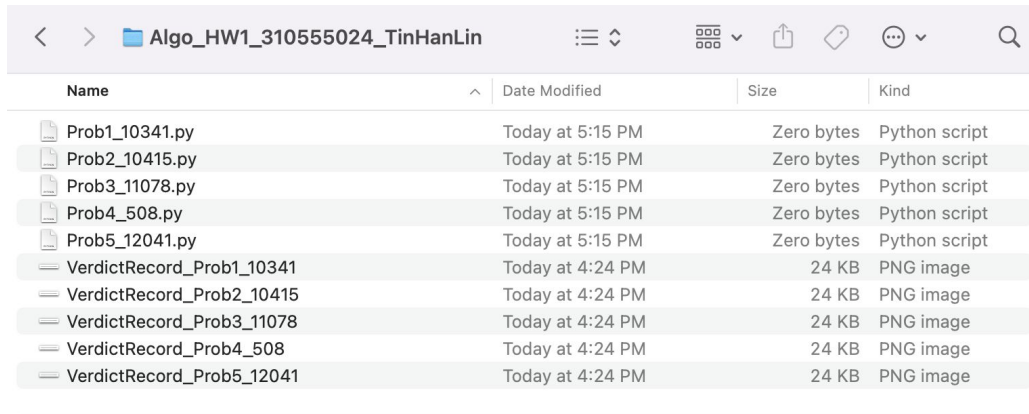
C++

0.280

2022-06-20
14:27:48

Uploading Rules

- You must upload **two things** for each problem:
 - Screenshot of Verdict Record
 - Source Code
- File Naming Rules**
→ Refer to the image on the right side 
- Name the zip file with “**Algo_HW1_yourStudentID_yourName.zip**”, e.g.
Algo_HW1_301555024_TinHanLin.zip
- Upload** the zip file to **E3 system**.



Name	Date Modified	Size	Kind
Prob1_10341.py	Today at 5:15 PM	Zero bytes	Python script
Prob2_10415.py	Today at 5:15 PM	Zero bytes	Python script
Prob3_11078.py	Today at 5:15 PM	Zero bytes	Python script
Prob4_508.py	Today at 5:15 PM	Zero bytes	Python script
Prob5_12041.py	Today at 5:15 PM	Zero bytes	Python script
VerdictRecord_Prob1_10341	Today at 4:24 PM	24 KB	PNG image
VerdictRecord_Prob2_10415	Today at 4:24 PM	24 KB	PNG image
VerdictRecord_Prob3_11078	Today at 4:24 PM	24 KB	PNG image
VerdictRecord_Prob4_508	Today at 4:24 PM	24 KB	PNG image
VerdictRecord_Prob5_12041	Today at 4:24 PM	24 KB	PNG image

Scoring Criteria

- Each Problem → 20 points (total score: $5 \times 20 = 100$)
If you pass the verdict of Online Judge and the program checking (clear programming and no plagiarism), you will get full points.
- Late turn-in will be subject to deduction of $1/4$, $1/3$, $1/2$ for one day, two days, and three days, respectively. That is, your score will be multiplied by $3/4$, $2/3$, and $1/2$ for being 1-day, 2-day, 3-day late. No turn-in allowed after 3 days.
- Warning: For the same group of plagiarism, the homework will be scored 0 points for all group members.

HW1 Intro and Important Dates

HW1 Introduction

- Chapters:
 - Others (based on basic programming skills only)
 - Divide-and-Conquer
- Difficulties:
 - Easy*2, Medium*2, Hard*1
- Platforms:
 - Online Judge
(<https://onlinejudge.org/>)
 - E3
(<https://e3.nycu.edu.tw/my/>)
- Programming Languages
→ C, C++, JAVA, Python, PASCAL

HW1 Important Dates

- **Release Date: 9/27 (Tue.) 00:01**
- **Due Date: 10/18 (Tue.) 23:59**
- Duration: 3 weeks
- **Note: Please start writing your homework as soon as possible, because Online Judge might be under maintenance from time to time.**

Homework 1 Problems (1/6) - Overview

1. [10341 - Solve It](#)
2. [10415 - Eb Alto Saxophone Player](#)
3. [11078 - Open Credit System](#)
4. [508 - Morse Mismatches](#)
5. [12041 - BFS \(Binary Fibonacci String\)](#)

Homework 1 Problems (2/6) - Prob1 / 10341 - Solve It

- Chapter: [Divide-and-conquer](#)
- Difficulty: [Easy](#)
- Problem Statement
 - Input
 - A sequence of coefficients of the equation
 - Output
 - Solution x to the equation
 - Objective
 - Minimize execution time
 - Constraints
 - Input scale
 - Input range

Homework 1 Problems (3/6) - Prob2 / 10415 - Eb Alto Saxophone Player

- Chapter: [Others \(based on basic programming skills only\)](#)
- Difficulty: [Easy](#)
- Problem Statement
 - Input
 - A sequence of characters (a song)
 - Output
 - 10 numbers indicating the number of presses for each finger
 - Objective
 - Minimize execution time
 - Constraints
 - Input scale
 - Input range
 - Compute the output based on rule given by problem

Homework 1 Problems (4/6) - Prob3 / 11078 - Open Credit System

- Chapter: [Others \(based on basic programming skills only\)](#)
- Difficulty: [Medium](#)
- Problem Statement
 - Input
 - Amount of students
 - A sequence of scores (for each student)
 - Output
 - The maximum amount of score that a senior student gets more than any junior student
 - Objective
 - Minimize execution time
 - Constraints
 - Input scale
 - Input range
 - In sequence of score, if $i < j$, then i 'th student is senior to the j 'th student

Homework 1 Problems (5/6) - Prob4 / 508 - Morse Mismatches

- Chapter: [Others \(based on basic programming skills only\)](#)
- Difficulty: [Medium](#)
- Problem Statement
 - Input
 - Morse code table
 - Context section
 - Morse words
 - Output
 - Matching word from context
 - Proper punctuation
 - Objective
 - Minimize execution time
 - Constraints
 - Input scale
 - Input range

Problem link: [508 - Morse Mismatches](#)

Homework 1 Problems (6/6) - Prob5 / 12041 - BFS (Binary Fibonacci String)

- Chapter: [Divide-and-conquer](#)
- Difficulty: [Hard](#)
- Problem Statement
 - Input
 - A number n (means $\text{Fib}(n)$)
 - Left index
 - Right index
 - Output
 - Content in $\text{Fib}(n)$ from left index to right index
 - Objective
 - Minimize execution time
 - Constraints
 - Input scale
 - Input range
 - Input relationship of the number

Problem link: [12041 - BFS \(Binary Fibonacci String\)](#)

HW2 Intro and Important Dates

HW2 Introduction

- Chapters:
 - Sorting
- Difficulties:
 - Easy*1, Medium*3, Hard*1
- Platforms:
 - Online Judge
(<https://onlinejudge.org/>)
 - E3
(<https://e3.nycu.edu.tw/my/>)
- Programming Languages
 - ➔ C, C++, JAVA, Python, PASCAL

HW2 Important Dates

- **Release Date: 10/18 (Tue.) 00:01**
- **Due Date: 11/8 (Tue.) 23:59**
- Duration: 3 weeks
- **Note: Please start writing your homework as soon as possible, because Online Judge might be under maintenance from time to time.**

Homework 2 Problems (1/6) - Overview

1. [11462 - Age Sort](#)
2. [10810 - Ultra-QuickSort](#)
3. [263 - Number Chains](#)
4. [482 - Permutation Arrays](#)
5. [110 - Meta-Loopless Sorts](#)

Homework 2 Problems (2/6) - Prob1 / 11462 - Age Sort

- Chapter: [Sorting](#)
- Difficulty: [Easy](#)
- Problem Statement
 - Input
 - Total number of people
 - n integers indicating the ages
 - Output
 - Sorted age sequence
 - Objective
 - Minimize execution time
 - Constraints
 - Input scale
 - Input range

Homework 2 Problems (3/6) - Prob2 / 10810 - Ultra-QuickSort

- Chapter: [Sorting](#)
- Difficulty: [Medium](#)
- Problem Statement
 - Input
 - Length of input sequence
 - A sequence of numbers
 - Output
 - Minimum number of swap operations
 - Objective
 - Minimize execution time
 - Constraints
 - Input scale
 - Input range

Homework 2 Problems (4/6) - Prob3 / 263 - Number Chains

- Chapter: [Sorting](#)
- Difficulty: [Medium](#)
- Problem Statement
 - Input
 - A sequence of positive numbers
 - Output
 - Number chains
 - Length of chain
 - Objective
 - Minimize execution time
 - Constraints
 - Input scale
 - Input range

Homework 2 Problems (5/6) - Prob4 / 482 - Permutation Arrays

- Chapter: [Sorting](#)
- Difficulty: [Medium](#)
- Problem Statement
 - Input
 - Two lines of numbers
 - Output
 - The list of floating numbers
 - Objective
 - Minimize execution time
 - Constraints
 - Input scale
 - Input range
 - Order the output based on the order of the first input line numbers

Homework 2 Problems (6/6) - Prob5 / 110 - Meta-Loopless Sorts

- Chapter: [Sorting](#)
- Difficulty: [Hard](#)
- Problem Statement
 - Input
 - An integer n
 - Output
 - Pascal code for outputting n numbers comparing size procedure
 - Objective
 - Minimize execution time
 - Constraints
 - Input scale
 - Input range
 - Meet the Pascal criteria in the problem description

HW3 Intro and Important Dates

HW3 Introduction

- Chapters:
 - Dynamic Programming
 - Greedy Algorithms
- Difficulties:
 - Easy*2, Medium*2, Hard*1
- Platforms:
 - Online Judge
(<https://onlinejudge.org/>)
 - E3
(<https://e3.nycu.edu.tw/my/>)
- Programming Languages
→ C, C++, JAVA, Python, PASCAL

HW3 Important Dates

- **Release Date: 11/8 (Tue.) 00:01**
- **Due Date: 11/29 (Tue.) 23:59**
- Duration: 3 weeks
- **Note: Please start writing your homework as soon as possible, because Online Judge might be under maintenance from time to time.**

Homework 3 Problems (1/6) - Overview

1. [10131 - Is Bigger Smarter?](#)
2. [10763 - Foreign Exchange](#)
3. [111 - History Grading](#)
4. [10440 - Ferry Loading II](#)
5. [709 - Formatting Text](#)

Homework 3 Problems (2/6) - Prob1 / 10131 - Is Bigger Smarter?

- Chapter: [Dynamic programming](#)
- Difficulty: [Easy](#)
- Problem Statement
 - Input
 - Information for each elephant (weight, and IQ)
 - Output
 - A sequence of elephant index
 - Objective
 - Minimize execution time
 - Constraints
 - Input scale
 - Input range
 - Output sequence should be the longest one (weight: low → high, IQ: high → low)

Problem link: [10131 - Is Bigger Smarter?](#)

Homework 3 Problems (3/6) - Prob2 / 10763 - Foreign Exchange

- Chapter: [Greedy algorithms](#)
- Difficulty: [Easy](#)
- Problem Statement
 - Input
 - Exchange information for each candidate (original location, and target location)
 - Output
 - YES or NO
(if there is a way for the exchange program to work out)
 - Objective
 - Minimize execution time
 - Constraints
 - Input scale
 - Input range

Homework 3 Problems (4/6) - Prob3 / 111 - History Grading

- Chapter: [Dynamic programming](#)
- Difficulty: [Medium](#)
- Problem Statement
 - Input
 - Number of events
 - Correct chronological order of n events
 - Student's chronological ordering of the n events
 - Output
 - Score for each student's ranking
 - Objective
 - Minimize execution time
 - Constraints
 - Input scale
 - Input range

Homework 3 Problems (5/6) - Prob4 / 10440 - Ferry Loading II

- Chapter: [Greedy algorithms](#)
- Difficulty: [Medium](#)
- Problem Statement
 - Input
 - Number of cars of a ferry across the river that can take (n)
 - Time for crossing the river (t) (same to returning time)
 - Number of cars waiting across the river (m)
 - Arrival time for each car
 - Output
 - Time of the last car is delivered to the other side of the river
 - Minimum number of trips made by the ferry to carry the cars within that time
 - Objective
 - Minimize execution time
 - Constraints
 - Input scale
 - Input range

Problem link: [10440 - Ferry Loading II](#)

Homework 3 Problems (6/6) - Prob5 / 709 - Formatting Text

- Chapter: [Dynamic programming](#)
- Difficulty: [Hard](#)
- Problem Statement
 - Input
 - A text consisting of several paragraphs
 - Desired width of the paragraph
 - Output
 - Formatted text
 - Objective
 - Minimize execution time
 - Constraints
 - Input scale
 - Input range
 - Minimize the sum of all badnesses
(badness of a gap of n spaces is $(n - 1)^2$)

Problem link: [709 - Formatting Text](#)

HW4 Intro and Important Dates

HW4 Introduction

- Chapters:
 - Graph Algorithms
- Difficulties:
 - Easy*1, Medium*1, Hard*3
- Platforms:
 - Online Judge
(<https://onlinejudge.org/>)
 - E3
(<https://e3.nycu.edu.tw/my/>)
- Programming Languages
 - ➔ C, C++, JAVA, Python, PASCAL

HW4 Important Dates

- **Release Date: 11/29 (Tue.) 00:01**
- **Due Date: 12/20 (Tue.) 23:59**
- Duration: 3 weeks
- **Note: Please start writing your homework as soon as possible, because Online Judge might be under maintenance from time to time.**

Homework 4 Problems (1/6) - Overview

1. [11045 - My T-shirt suits me](#)
2. [1504 - Genghis Khan the Conqueror](#)
3. [1056 - Degrees of Separation](#)
4. [11747 - Heavy Cycle Edges](#)
5. [12797 - Letters](#)

Homework 4 Problems (2/6) - Prob1 / 11045 - My T-shirt suits me

- Chapter: [Graph algorithms](#)
- Difficulty: [Easy](#)
- Problem Statement
 - Input
 - Number of T-shirts
 - Number of volunteers
 - Two sizes of T-shirts suit each volunteer
 - Output
 - If Victor can distribute T-shirts in such a way that all volunteers get a T-shirt that suit them
 - Objective
 - Minimize execution time
 - Constraints
 - Input scale
 - Input range

Problem link: [11045 - My T-shirt suits me](#)

Homework 4 Problems (3/6) - Prob2 / 1504 - Genghis Khan the Conqueror

- Chapter: [Graph algorithms](#)
- Difficulty: [Medium](#)
- Problem Statement
 - Input
 - Number of cities
 - Roads in Pushtuar
 - Costs of bidirectional road
 - Suspicious road cost changes
 - Output
 - Expected minimal total cost
 - Objective
 - Minimize execution time
 - Constraints
 - Input scale
 - Input range

Problem link: [1504 - Genghis Khan the Conqueror](#)

Homework 4 Problems (4/6) - Prob3 / 1056 - Degrees of Separation

- Chapter: [Graph algorithms](#)
- Difficulty: [Hard](#)
- Problem Statement
 - Input
 - Number of people in the network
 - Number of network relationships
 - Relationships
 - Output
 - Maximum degree of separation
 - Objective
 - Minimize execution time
 - Constraints
 - Input scale
 - Input range

Homework 4 Problems (5/6) - Prob4 / 11747 - Heavy Cycle Edges

- Chapter: [Graph algorithms](#)
- Difficulty: [Hard](#)
- Problem Statement
 - Input
 - Number of nodes
 - Number of edges
 - Weights of edge
 - Output
 - Weights of all edges that are the heaviest edge in some cycle
 - Objective
 - Minimize execution time
 - Constraints
 - Input scale
 - Input range

Homework 4 Problems (6/6) - Prob5 / 12797 - Letters

- Chapter: [Graph algorithms](#)
- Difficulty: [Hard](#)
- Problem Statement
 - Input
 - Size of the park
 - Letters for defining the park
 - Output
 - Length of a shortest consistent path (For example, if they step over a lowercase c, they will not allow themselves stepping over an uppercase C afterwards.)
 - Objective
 - Minimize execution time
 - Constraints
 - Input scale
 - Input range