Home(/)

0x02. Minimum Operations

Algorithm Python

My Planning(/planning/me)

Weight: 1

Projects(/projects/current)

Project over - took place from Jul 8, 2024 6:00 AM to Jul 12, 2024 6:00 AM

An auto QA review will be launched at the deadline

QA reviews I can make(/corrections/to_review)

Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)

In a nutshell…

- **Auto QA review:** 12.0/12 mandatory
- **Altogether: 100.0%**
  - Mandatory: 100.0%
  - Optional: no optional tasks

Curriculums(/dashboards/my_curriculums)
Concepts(/concepts)
Conference rooms(/dashboards/video_rooms)
Servers(/servers)
Sandboxes(/user_containers/current)
Tools(/dashboards/my_tools)
Video on demand(/dashboards/videos)
Peers(/users/peers)
Discord(https://discord.com/app)

For this project, you will need to understand several key algorithmic and mathematical concepts to devise a solution that efficiently calculates the minimum number of operations to achieve a given number of characters using only “Copy All” and “Paste” operations. Here is a list of concepts and resources that will be helpful:

Concepts Needed:

1. **Dynamic Programming**:
   - Familiarity with dynamic programming can help in breaking down the problem into simpler subproblems and building up the solution.
   - Dynamic Programming (GeeksforGeeks) (/rltoken/l3JYgicNQw2Ue1Kg9jV80Q)
2. **Prime Factorization**:
   - Understanding how to perform prime factorization is crucial since the problem can be reduced to finding the sum of the prime factors of the target number n.
   - Prime Factorization (Khan Academy) (/rltoken/cFcADpVYRCl5pdut-Lemmg)
3. **Code Optimization**:
   - Knowing how to approach problems from an optimization perspective can be useful in finding the most efficient solution.
   - How to optimize Python code (/rltoken/98ZF5bRckUKror6pGJQlHQ)
4. **Greedy Algorithms**:
   - The problem can also be approached with greedy algorithms, choosing the best option at each step.
   - Greedy Algorithms (GeeksforGeeks) (/rltoken/k6-mba0b4nayJi0VqYhKjQ)
5. **Basic Python Programming**:

My Profile(/users/my_profile)

- Proficiency in Python, including loops, conditionals, and functions, is necessary to implement the solution.
- Python Functions (Python Official Documentation) (/rltoken/ao3SJVl4yY1SfugfVa3anw)

By studying these concepts and utilizing the resources provided, you will be equipped to tackle the "Minimum Operations" problem effectively, applying both mathematical reasoning and programming skills to find the most efficient solution.

Additional Resources

- Mock Technical Interview (/rltoken/HX0vuVl1V-9T4vvh8NDCyw)

Requirements

General

- Allowed editors: `vi`, `vim`, `emacs`
- All your files will be interpreted/compiled on Ubuntu 20.04 LTS using `python3` (version 3.4.3)
- All your files should end with a new line
- The first line of all your files should be exactly `#!/usr/bin/python3`
- A `README.md` file, at the root of the folder of the project, is mandatory
- Your code should be documented
- Your code should use the `PEP 8` style (version 1.7.x)
- All your files must be executable

Tasks

0. Minimum Operations

mandatory

Score: 100.0% (*Checks completed: 100.0%*)

In a text file, there is a single character `H`. Your text editor can execute only two operations in this file: `Copy All` and `Paste`. Given a number `n`, write a method that calculates the fewest number of operations needed to result in exactly `n` `H` characters in the file.

- Prototype: `def minOperations(n)`
- Returns an integer
- If `n` is impossible to achieve, return `0`

Example:

`n = 9`

```
H => Copy All => Paste => HH => Paste => HHH => Copy All => Paste => HHHHHH => Paste
=> HHHHHHHHH
```

Number of operations: `6`

```
carrie@ubuntu:~/0x02-minoperations$ cat 0-main.py
#!/usr/bin/python3
"""
Main file for testing
"""

minOperations = __import__('0-minoperations').minOperations

n = 4
print("Min # of operations to reach {} char: {}".format(n, minOperations(n)))

n = 12
print("Min # of operations to reach {} char: {}".format(n, minOperations(n)))

carrie@ubuntu:~/0x02-minoperations$
```

```
carrie@ubuntu:~/0x02-minoperations$ ./0-main.py
Min number of operations to reach 4 characters: 4
Min number of operations to reach 12 characters: 7
carrie@ubuntu:~/0x02-minoperations$
```

Repo:

- GitHub repository: `alx-interview`
- Directory: `0x02-minimum_operations`
- File: `0-minoperations.py`

Check submission | View results