

Limited Factoring

0x00 General Overview

The aim of this test is to solve the following problem as documented at <https://gist.github.com/jcampbell-ipsi/1cca1c94f84fd7d198bda41b33f01a51>

Given a set of potential factors, find the shortest (by number of factors, then numerical order) path to the target number.

Folder structure

- **README.pdf**: The report including all documentation for this challenge
- **factoring.py**: The python solution for this challenge
- **test_factoring.py**: The test for factoring.py
- **requirements.txt**: Required libs for the solution and test

0x01 Instructions for Solution

Install requirements

```
pip install -r requirements.txt
```

Run solution

```
python factoring.py
```

Then, please follow the input constraints strictly to get the results.

Run test

```
pytest
```

0x02 Solution Designs

There are generally 3 parts in the `factoring.py`, which are input validation, get shortlisted factors, and get shortest path.

Input Validation

The input validation strictly limits the input values are following the constraints.

- Target number (N) can only be positive integer less than 2^{60}
- Number of factors (K) is a positive integer less than 32
- Possible factors (A) is a set of unique integers, each less than 2^{60} , with K elements

If the values are out of range, or some of the possible factors are duplicate, or the number of possible factors not equal to K, the input will be rejected.

Get shortlisted factors

This is a filter for the possible factors and prepare the useful factors for the shortest path.

To get the shortest path, it must use large factors as much as possible. So, the shortlist is sorted descending.

The numbers which cannot be fully divided by the target is also not usable for the calculation, so just trim them out to reduce the loop.

Get shortest path

This is the core search function and designed as recursive function to get the shortest when the factors can be divided by the target number.

And the return value builds up the whole path when it reaches 1, the end of the recursive.