

Problem Solving Template (change the title)

To learn how to use this template, check out the course ["Data Structures and Algorithms in Python"](#).

How to run the code and save your work

The recommended way to run this notebook is to click the "Run" button at the top of this page, and select "Run on Binder". This will run the notebook on mybinder.org, a free online service for running Jupyter notebooks.

This tutorial is an executable [Jupyter notebook](#). You can *run* this tutorial and experiment with the code examples in a couple of ways: *using free online resources* (recommended) or *on your computer*.

Option 1: Running using free online resources (1-click, recommended)

The easiest way to start executing the code is to click the **Run** button at the top of this page and select **Run on Binder**. You can also select "Run on Colab" or "Run on Kaggle", but you'll need to create an account on [Google Colab](#) or [Kaggle](#) to use these platforms.

Option 2: Running on your computer locally

To run the code on your computer locally, you'll need to set up [Python](#), download the notebook and install the required libraries. We recommend using the [Conda](#) distribution of Python. Click the **Run** button at the top of this page, select the **Run Locally** option, and follow the instructions.

Saving your work

Before starting the assignment, let's save a snapshot of the assignment to your [Jovian](#) profile, so that you can access it later, and continue your work.

```
project_name = ??? # give it an appropriate name
```

```
!pip install jovian --upgrade --quiet
```

```
import jovian
```

```
jovian.commit(project=project_name)
```

```
[jovian] Attempting to save notebook..
```

```
[jovian] Updating notebook "aakashns/python-problem-solving-template" on  
https://jovian.ai/
```

```
[jovian] Uploading notebook..
```

```
[jovian] Capturing environment..
```

```
[jovian] Committed successfully! https://jovian.ai/aakashns/python-problem-solving-template
```

```
'https://jovian.ai/aakashns/python-problem-solving-template'
```

Problem Statement

??? (past the problem statement here)

Source: ??? (link to the source)

The Method

Here's the systematic strategy we'll apply for solving problems:

1. State the problem clearly. Identify the input & output formats.
2. Come up with some example inputs & outputs. Try to cover all edge cases.
3. Come up with a correct solution for the problem. State it in plain English.
4. Implement the solution and test it using example inputs. Fix bugs, if any.
5. Analyze the algorithm's complexity and identify inefficiencies, if any.
6. Apply the right technique to overcome the inefficiency. Repeat steps 3 to 6.

This approach is explained in detail in [Lesson 1](#) of the course. Let's apply this approach step-by-step.

Solution

1. State the problem clearly. Identify the input & output formats.

While this problem is stated clearly enough, it's always useful to try and express in your own words, in a way that makes it most clear for you.

Problem

??? (express the problem clearly in our own words, and in abstract terms)

Input

1. ???
2. ???

(add more if required)

Output

1. ???

(add more if required)

Based on the above, we can now create a signature of our function:

Create a function signature here. The body of the function can contain a single state

Save and upload your work before continuing.

```
import jovian
```

```
jovian.commit()
```

[jovian] Attempting to save notebook..

2. Come up with some example inputs & outputs. Try to cover all edge cases.

Our function should be able to handle any set of valid inputs we pass into it. Here's a list of some possible variations we might encounter:

1. ???

2. ???

3. ???

4. ???

5. ???

(add more if required)

We'll express our test cases as dictionaries, to test them easily. Each dictionary will contain 2 keys: `input` (a dictionary itself containing one key for each argument to the function and `output` (the expected result from the function).

```
test = {  
    'input': {  
        ???  
    },  
    'output': ???  
}
```

Create one test case for each of the scenarios listed above. We'll store our test cases in an array called `tests`.

```
tests = []
```

```
tests.append(test)
```

```
tests.append(  
    'input': {  
        ???  
    },  
    'output': ???  
)
```

```
# add more test cases
```

3. Come up with a correct solution for the problem. State it in plain English.

Our first goal should always be to come up with a *correct* solution to the problem, which may not necessarily be the most *efficient* solution. Come with a correct solution and explain it in simple words below:

1. ???
2. ???
3. ???
4. ???
5. ???

(add more steps if required)

Let's save and upload our work before continuing.

```
jovian.commit()
```

```
[jovian] Attempting to save notebook..
```

```
[jovian] Updating notebook "aakashns/python-problem-solving-template" on
```

```
https://jovian.ai/
```

```
[jovian] Uploading notebook..
```

```
[jovian] Capturing environment..
```

```
[jovian] Committed successfully! https://jovian.ai/aakashns/python-problem-solving-template
```

```
'https://jovian.ai/aakashns/python-problem-solving-template'
```

4. Implement the solution and test it using example inputs. Fix bugs, if any.

We can test the function by passing the input to it directly or by using the `evaluate_test_case` function from `jovian`.

```
from jovian.pythondsa import evaluate_test_case
```

```
evaluate_test_case(???)
```

Evaluate your function against all the test cases together using the `evaluate_test_cases` (plural) function from `jovian`.

```
from jovian.pythondsa import evaluate_test_cases
```

```
evaluate_test_cases(???)
```

Verify that all the test cases were evaluated. We expect them all to fail, since we haven't implemented the function yet.

Let's save our work before continuing.

```
jovian.commit()
```

```
[jovian] Attempting to save notebook..
```

```
[jovian] Updating notebook "aakashns/python-problem-solving-template" on
```

```
https://jovian.ai/
```

```
[jovian] Uploading notebook..
```

```
[jovian] Capturing environment..
```

```
[jovian] Committed successfully! https://jovian.ai/aakashns/python-problem-solving-template
```

```
'https://jovian.ai/aakashns/python-problem-solving-template'
```

5. Analyze the algorithm's complexity and identify inefficiencies, if any.

```
jovian.commit()
```

```
[jovian] Attempting to save notebook..
```

```
[jovian] Updating notebook "aakashns/python-problem-solving-template" on
```

```
https://jovian.ai/
```

```
[jovian] Uploading notebook..
```

```
[jovian] Capturing environment..
```

```
[jovian] Committed successfully! https://jovian.ai/aakashns/python-problem-solving-template
```

```
'https://jovian.ai/aakashns/python-problem-solving-template'
```

6. Apply the right technique to overcome the inefficiency. Repeat steps 3 to 6.

```
jovian.commit()
```

```
[jovian] Attempting to save notebook..
```

```
[jovian] Updating notebook "aakashns/python-problem-solving-template" on
```

```
https://jovian.ai/
```

```
[jovian] Uploading notebook..
```

```
[jovian] Capturing environment..
```

```
[jovian] Committed successfully! https://jovian.ai/aakashns/python-problem-solving-template
```

```
'https://jovian.ai/aakashns/python-problem-solving-template'
```

7. Come up with a correct solution for the problem. State it in plain English.

Come with the optimized correct solution and explain it in simple words below:

1. ???

2. ???

3. ???

4. ???

5. ???

(add more steps if required)

Let's save and upload our work before continuing.

```
jovian.commit()
```

```
[jovian] Attempting to save notebook..
```

```
[jovian] Updating notebook "aakashns/python-problem-solving-template" on
```

```
https://jovian.ai/
```

```
[jovian] Uploading notebook..
```

```
[jovian] Capturing environment..
```

```
[jovian] Committed successfully! https://jovian.ai/aakashns/python-problem-solving-template
```

```
'https://jovian.ai/aakashns/python-problem-solving-template'
```

8. Implement the solution and test it using example inputs. Fix bugs, if any.

9. Analyze the algorithm's complexity and identify inefficiencies, if any.

If you found the problem on an external platform, you can make a submission to test your solution.

Share your approach and start a discussion on the Jovian forum: <https://jovian.ai/forum/c/data-structures-and-algorithms-in-python/78>

jovian.commit()

[jovian] Attempting to save notebook..