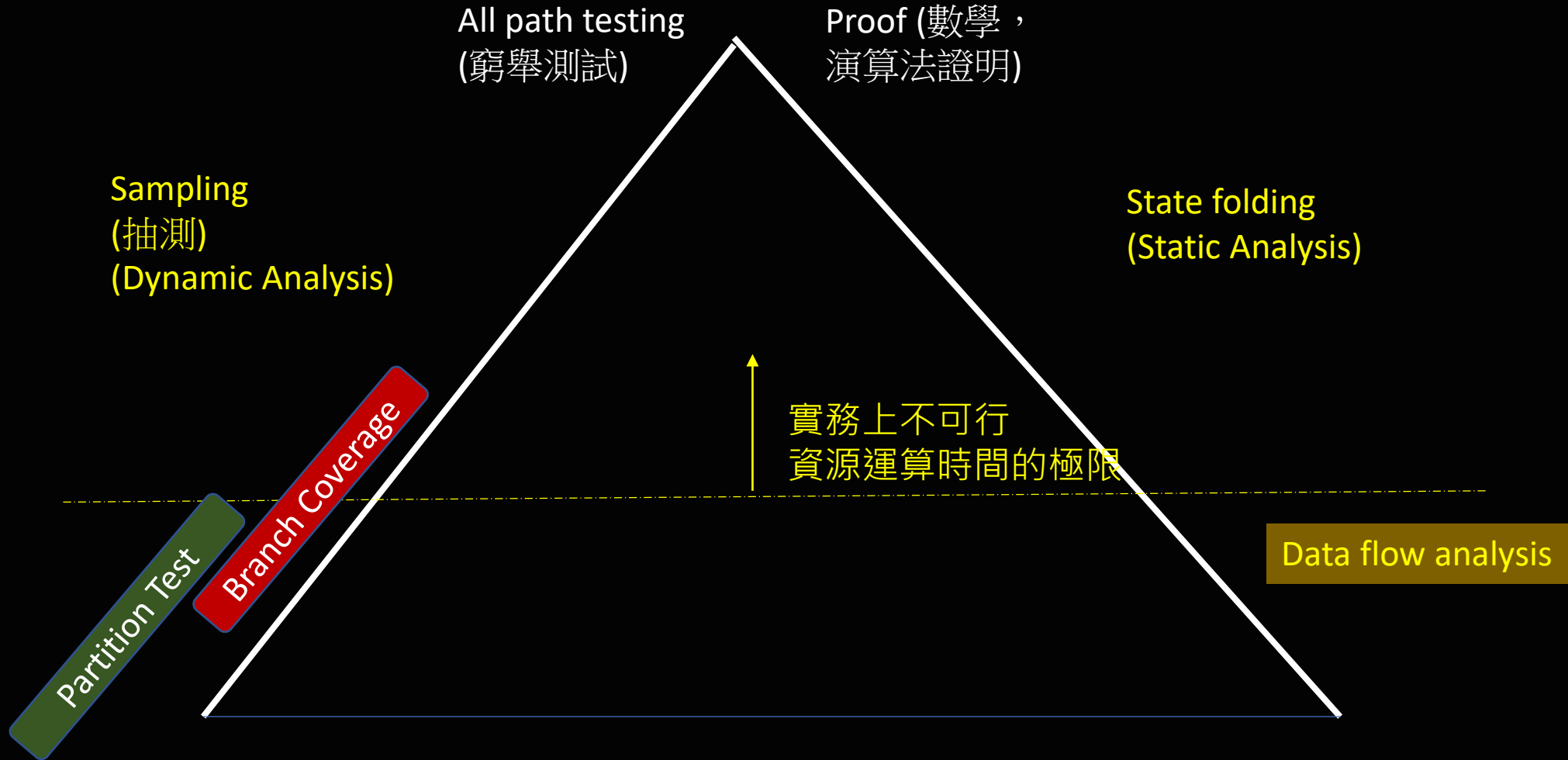
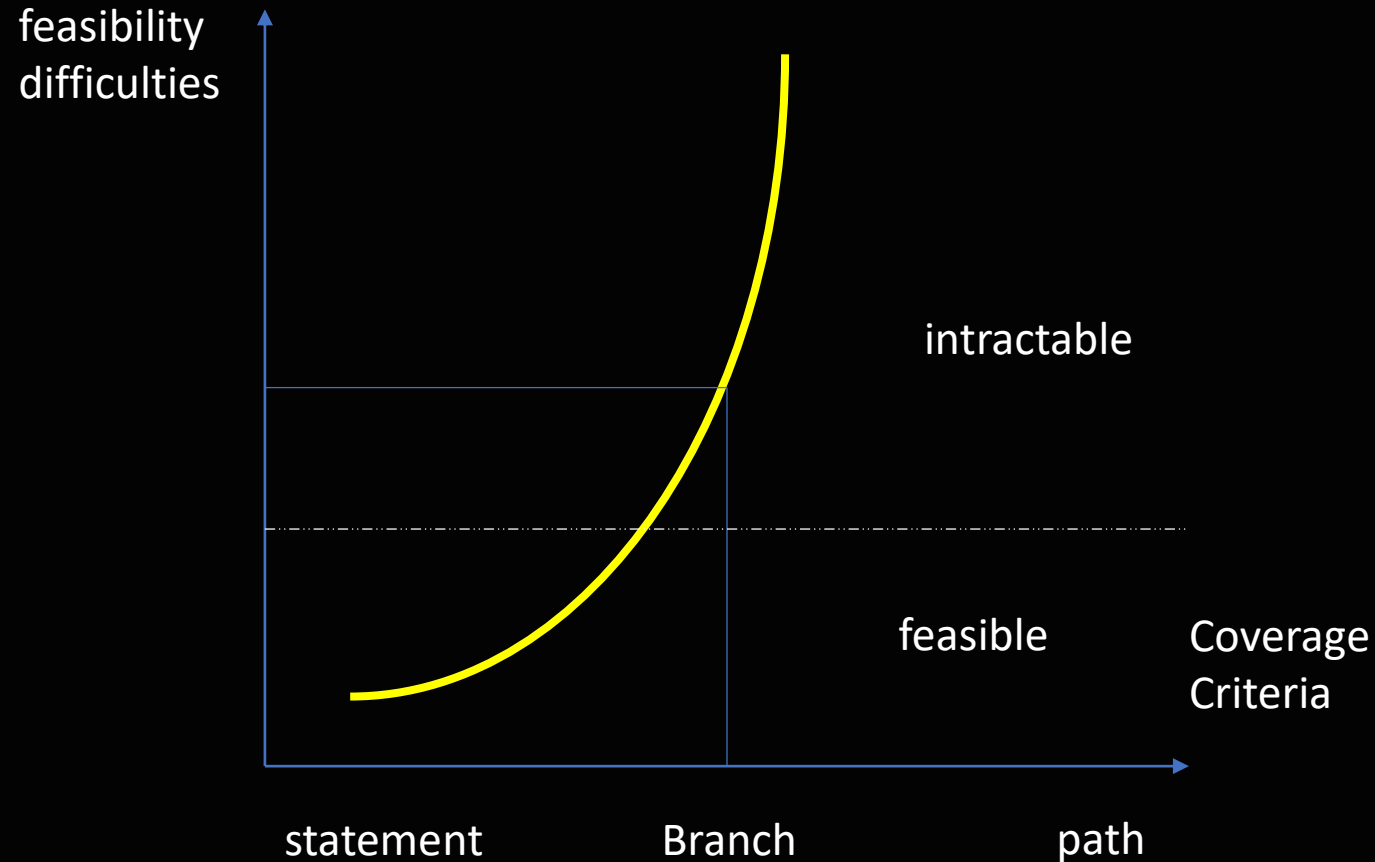


Loop Testing



Recall about branch/path Coverage

- 理論上非常有趣，你如果想達到的標準是 branch coverage，實務上他已經很困難了
- 聽說過 branch coverage 是金管會證交系統的驗收標準？
- 總不能說迴圈測試太難，所以就不用做了吧？



Why Loop Testing

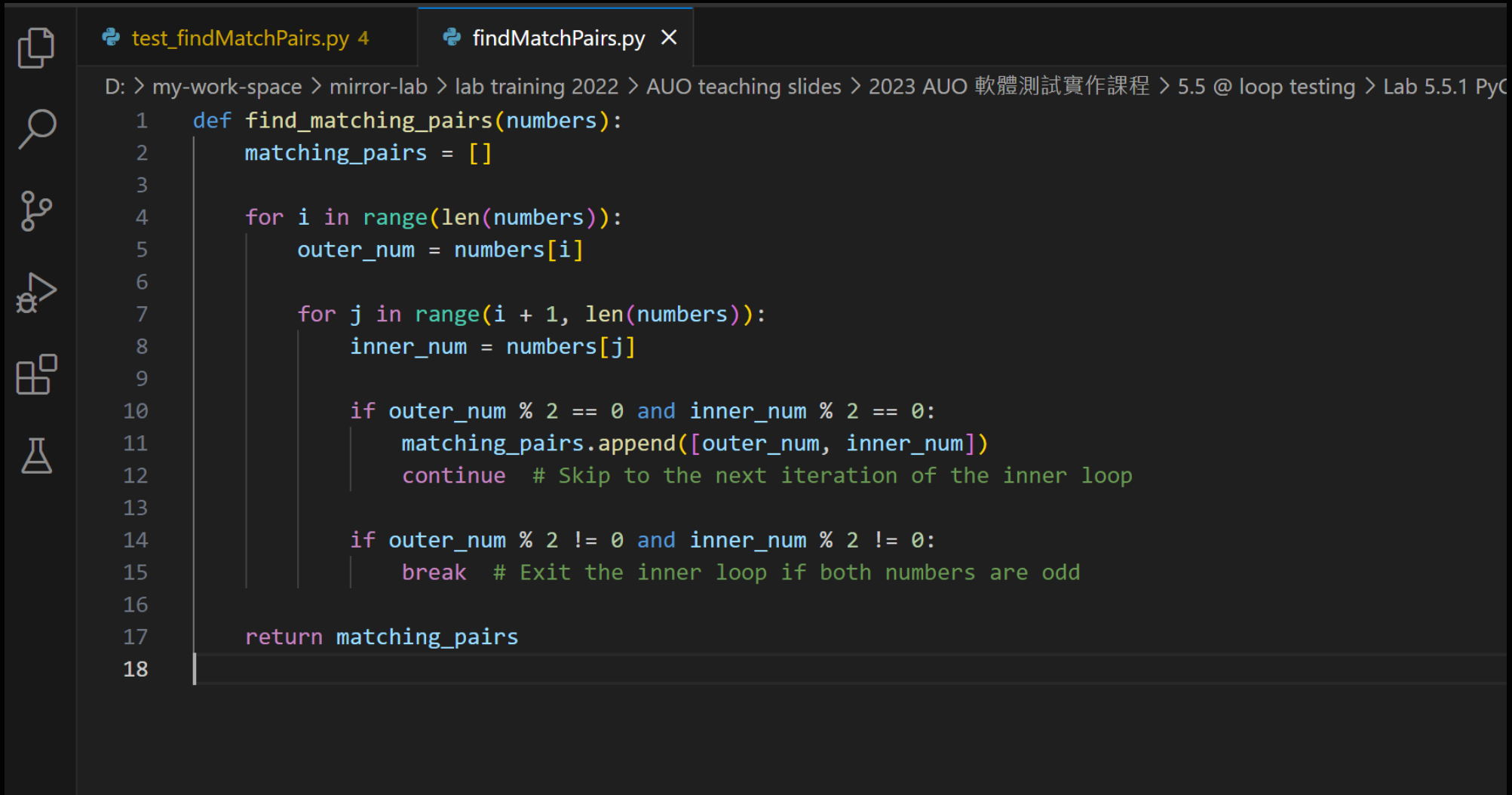
- 程式中總是會牽涉到演算法，以及各種商業運算用到 loop
- 一旦程式有了 Loop 測試的難度往往提升到 path coverage
- 你不太可能會真正做到 path coverage
- 但是你應該注意到如何在有限的代價之下進行應該要做的測試
- 這就是 Loop testing

Loop testing 關注的重點

- 迴圈測試 (Loop Testing) 是一種軟體測試技術，用於驗證程式中的迴圈結構是否能正確運作。在迴圈測試中，目標是檢測迴圈的不同路徑、邊界條件和特殊情況，以確保迴圈能夠正確執行並處理各種情況。
- 以下是一些迴圈測試的內容和技巧：
- **迴圈結構**：首先，測試迴圈的基本結構，包括迴圈的進入點、結束條件、迭代條件和迴圈內的執行邏輯。確保迴圈能夠正確運作，並在達到結束條件時停止執行。
- **迴圈次數**：測試迴圈執行的次數，包括迴圈執行的最小次數、最大次數和一般情況下的次數。這有助於驗證迴圈的迭代次數是否符合預期。
- **迴圈邊界條件**：測試迴圈的邊界條件，例如邊界值、邊界條件的上下限和邊界條件的變化情況。這有助於驗證迴圈在邊界條件下的行為是否正確。
- **迴圈內部路徑**：測試迴圈內部的不同路徑，包括迴圈內的條件判斷、分支和迴圈終止條件的變化。這可以確保迴圈在各種情況下都能正確運作。
- **特殊情況**：測試特殊情況下的迴圈行為，例如迴圈初始化、空迴圈和無窮迴圈的情況。這有助於驗證迴圈在特殊情況下是否能正確處理。
- **迴圈的複雜性**：考慮迴圈的複雜性，例如巢狀迴圈、迴圈中的控制流程和迴圈的中斷條件。這有助於驗證迴圈在複雜情況下的正確性。

**Lab 5.5.1: 請下載
findMatchPairs.js 並且寫出你
心中想像的 Loop test cases.**

先幫各位講解一下程式碼在做什麼






The image shows a Visual Studio Code editor window. On the left is a sidebar with icons for Explorer, Search, Source Control, Run and Debug, Extensions, and Testing. The Explorer panel shows two files: `test_findMatchPairs.py` and `findMatchPairs.py`. The `findMatchPairs.py` file is open in the editor, showing the following Python code:

```
D: > my-work-space > mirror-lab > lab training 2022 > AUO teaching slides > 2023 AUO 軟體測試實作課程 > 5.5 @ loop testing > Lab 5.5.1 PyC

1  def find_matching_pairs(numbers):
2      matching_pairs = []
3
4      for i in range(len(numbers)):
5          outer_num = numbers[i]
6
7          for j in range(i + 1, len(numbers)):
8              inner_num = numbers[j]
9
10             if outer_num % 2 == 0 and inner_num % 2 == 0:
11                 matching_pairs.append([outer_num, inner_num])
12                 continue # Skip to the next iteration of the inner loop
13
14             if outer_num % 2 != 0 and inner_num % 2 != 0:
15                 break # Exit the inner loop if both numbers are odd
16
17     return matching_pairs
18
```

Discussion (隱藏)

 test_findMatchPairs.py 4 X findMatchPairs.pyk-space > mirror-lab > lab training 2022 > AUO teaching slides > 2023 AUO 軟體測試實作課程 > 5.5 @ loop testing > Lab 5.5.1 PyCode loop testing > answer >  te

```
1  def test_findMatchingPairs():
2      numbers1 = [1, 2, 3, 4, 5, 6]
3      pairs1 = find_matching_pairs(numbers1)
4      assert pairs1 == [[2, 4], [2, 6], [4, 6]]
5
6      numbers2 = [1, 3, 5, 7]
7      pairs2 = find_matching_pairs(numbers2)
8      assert pairs2 == []
9
10     numbers3 = [1, 3, 5, 7, 9]
11     pairs3 = find_matching_pairs(numbers3)
12     assert pairs3 == []
13
14     numbers4 = [2, 4, 6, 8, 10]
15     pairs4 = find_matching_pairs(numbers4)
16     assert pairs4 == [[2, 4], [2, 6], [2, 8], [2, 10], [4, 6], [4, 8], [4, 10], [6, 8], [6, 10], [8, 10]]
17
```