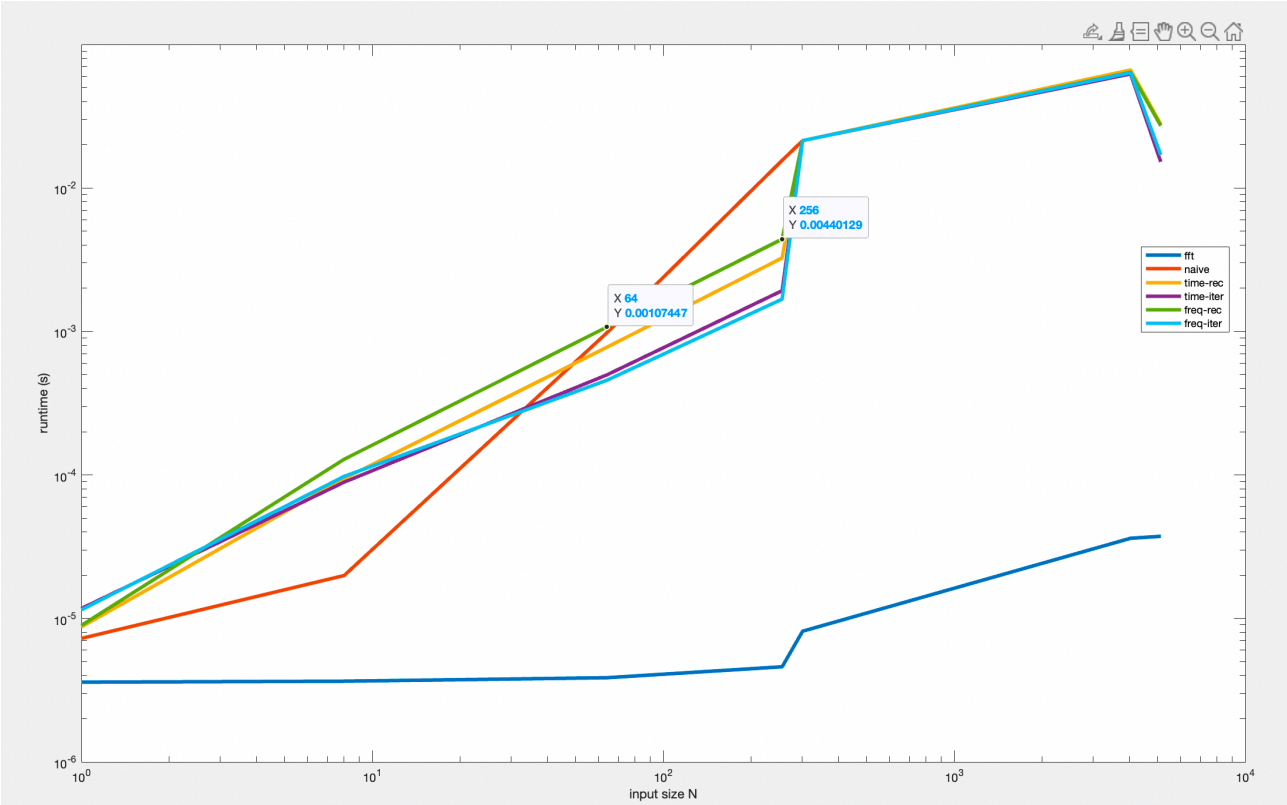
Part 1: 1–D DFT

1. Modify test 1d.m if needed and run it. Attach the generated plots.



2. Observe the run time of various algorithms under different input sizes. Explain.

已知輸入的大小為 n = [1, 8, 64, 256, 301, 4032, 5120]

當 n = 1 時

速度排序是 naive < time_rec = freq_rec < time_iter = freq_iter

原因是無論是 time 還是 freq 、 rec 還是 iter 都需要做一些處理，並且最後都要使用 naive 所以當然更慢一些

而 rec 比 iter 更快是因為 iter 需要額外算輸入有多少 2 的冪次。

並且， time 跟 freq 時間差不多是因為 n 就是 1 ，並不會額外做什麼處理，直接帶入 naive 。

當 n = 8 時

或許，因為 n 還是太小了，所以時間還是被程式架構給支配，rec 跟 iter 還是比 naive 慢，不過可以觀察到的是，iter 已經比 rec 還要快了。這很合理，因為 iter 本來效率就比 rec 好。

當 n = 64 時

除了 freq_rec ，其餘 fft 皆比 native 快了。但其實差不多。

當 n = 256 時

速度排序是 naive > freq_rec > time_rec > freq_iter > time_iter 。

當 n = 301 時

fft 們的速度突然變慢，原因很明顯，是因為 n 是奇數，所以速度等於 native 的速度。

當 n = 4032 與 5120 時

我們觀察到 5120 明顯快很多，我們可以觀察4032與5120中有多少個2：

```
>> dec2bin(4032)

ans =

     '111111000000'
```

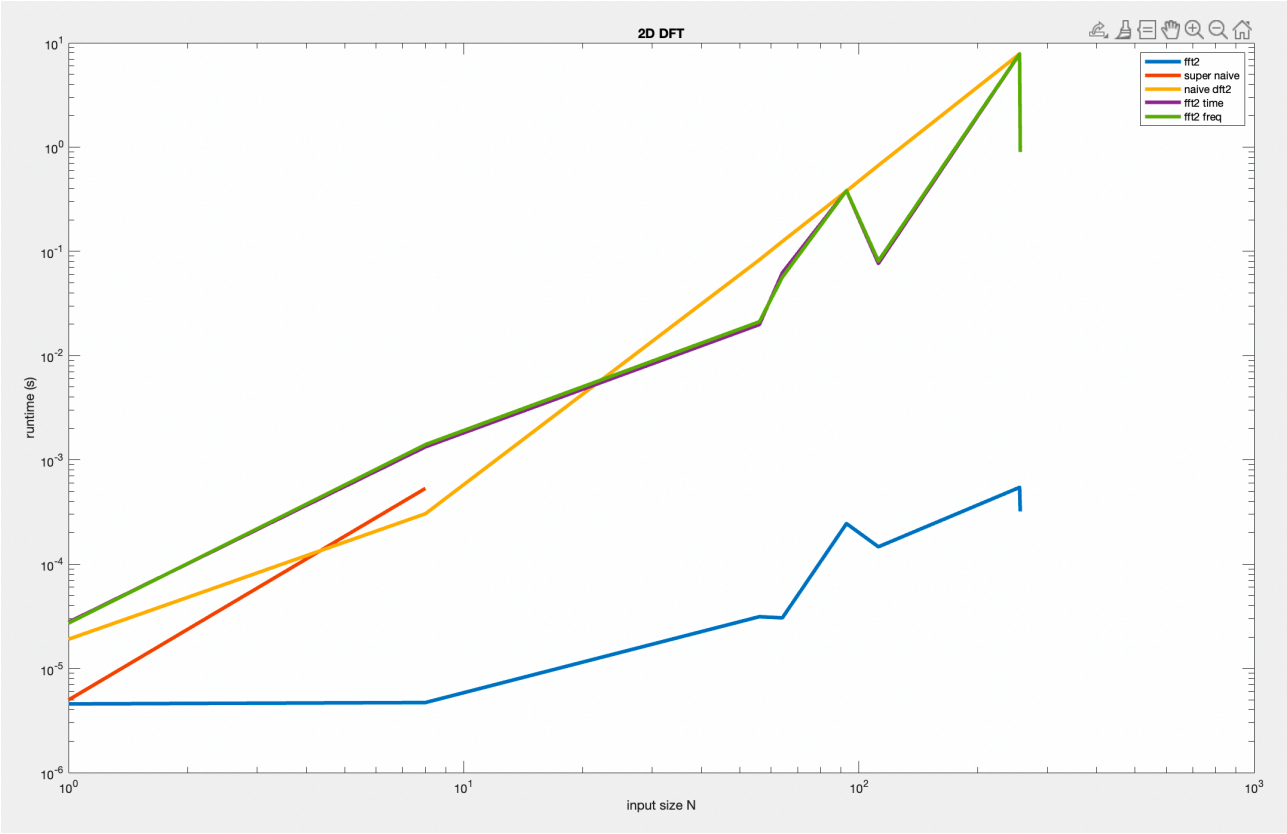
```
>> dec2bin(5120)

ans =

     '1010000000000'
```

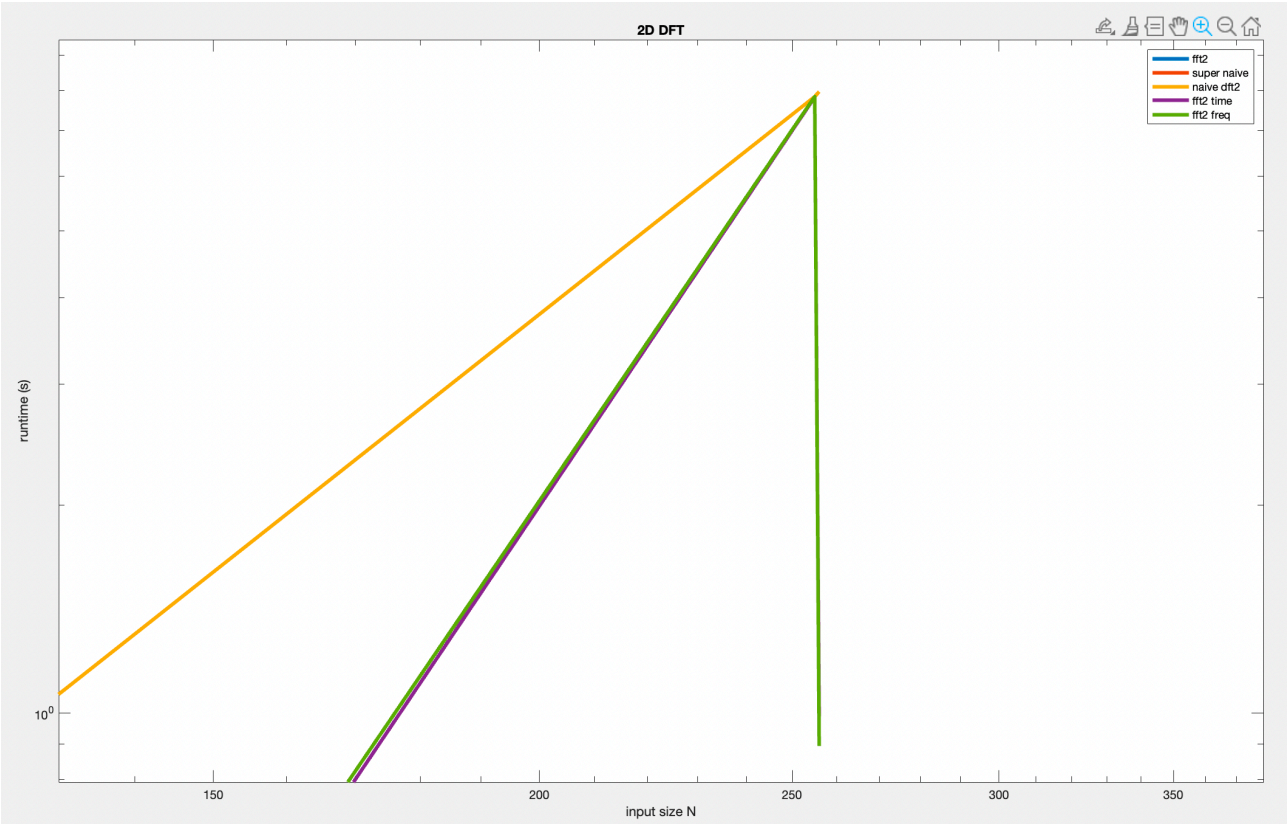
4032 有 6 個 2 ，而 5120 有 10 個 2。
還可以觀察到 4032 拿掉 2 我們要做的 dft 長度是 111111_2;
5120 拿掉 2 我們要做的 dft 長度是 101_2。
因為5120實際需要做的 dft 長度比 4032 的少太多了，所以 5120 速度比較快是非常合理的。

Part 2: 2-D DFT

Experiments [10]

1. Modify test 2d.m if needed and run it. Attach the generated plots.





When n =255,256

2. Observe the run time of various algorithms under different input sizes. Explain.
已知輸入的大小為 n = [1, 8, 56, 64, 93, 112, 255, 256];
這次，我從後面講回來。
很明顯，（255 , 256）、（93 , 112）、（56 , 64）是兩兩一組的。

當 n =（255 , 256）時
因為 255 是奇數，所以 my_fft2 速度等於 fft2_native 的速度。
而因 256 是 2 的冪次，所以速度突然加速，非常合理。

當 n =（93 , 112）時
因為 93 是奇數，所以 my_fft2 速度等於 fft2_native 的速度。
112 是偶數，所以速度較快。

比較有趣是 n =（56 , 64）時
在內建的 fft2 中 64 的速度較快的，但我們自己寫的 my_fft2 速度反而速度下降。
56 = 8*7
64 = 8*8
我猜想，這是老師故意設計的數字， 7 ≈ 8 ，看回到Part 1: 1–D DFT n=8時。
在當時， fft 就比 naive 來得慢，所以這裡速度下降，我覺得是非常合理的。

那當 n≤8 時，更不用說，肯定 fft2_native 比 my_fft2 來得快。
而 super_native 在 n = 1 時比較快，是因為他本身沒有多餘的操作，所以本來就比較快，但在n = 8 時就比 fft2_native 來得慢了，但還是因為 n 太小，時間被程式架構給支配，my_fft2 在此時仍比 super_native 漫，不過可以預期後面很快就會超過他了。

Questions

Answer all the following questions as detailed as possible.

1. [5] What does function check_error do?

在 test_1d 還有 test_2d 裡，會隨機生成 num_test 個複數矩陣，帶入盡我們的 fft 中測試效能。並且取出每個 fft 的第一筆測試資料，將內建的 fft 作為標準，測量每個 fft 的 error 。並且在error ≥ 1e-6時報錯。

輸入 ref (內建的 fft 測試輸出資料)、 x(某個 function 測試輸出資料)
沒有輸出，但在error ≥ 1e-6時報錯。

2. [5] What does function measure runtime do?

量測目標函數的執行時間，並且算出平均值與標準差(雖然標準差並未被使用)。

輸入 fun (某個 function)、 x(function的測試用資料,格式：an array of cell)
輸出 平均執行時間與執行時間標準差。

3. [5] What does script test 1d do?

t_{%function} 用來儲存每次的時間，以便最後畫圖用
num_test 用來設定測試幾次。

```
test_cases = cell(1, num_test);
for i = 1: num_test
    test_cases{1, i} = rand(n(k), 1) + j * rand(n(k), 1);
    if randi(2) == 1
        test_cases{1, i} = test_cases{1, i}.';
    end
end
```

這部分在做隨機生成測試資料，並且隨機轉直或轉橫。

之後進行測量時間，並且記住第一次的測試資料。
測量玩時間後用記住的測試資料量測誤差。

重複執行固定次數後，最後畫出圖形。

4. [5] What does script test 2d do?
t_{%function} 用來儲存每次的時間，以便最後畫圖用
num_test 用來設定測試幾次。

```
x = cell(1, num_test);
for i = 1: num_test
    x{1, i} = rand(n(k)) + j * rand(n(k));
end
```

這部分在做隨機生成測試資料，並且隨機轉直或轉橫。

之後進行測量時間，並且記住第一次的測試資料。
測量玩時間後用記住的測試資料量測誤差。

重複執行固定次數後，最後畫出圖形。