ชื่อ – นามสกุล .....รหัสนักศึกษา.....

**Data Structures and Algorithm**

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

**การทดลองที่ 3 :** Algorithm Analysis

**จุดประสงค์**

1. สามารถประเมิน Complexity ในโปรแกรมต่างๆ ได้

**ตอนที่ 1 :** Time Calculation

1. ให้นักศึกษาค้นหา Source Code ใน internet **ที่มีการทำงานเดียวกัน** 2 โปรแกรม แล้วใช้ฟังก์ชั่น time() เพื่อตรวจสอบกว่าการทำงานของทั้งสองโปรแกรม มีการใช้เวลาที่แตกต่างกันอย่างไรบ้าง

```
import time

st = time.time()
# source code
en = time.time()
elapsed_time = et - st
print('Execution time:', elapsed_time, 'seconds')
```

2. ให้นักศึกษาทดสอบการทำงานของ ฟังก์ชั่น NearestNeighbors ของ sklearn.neighbors เมื่อมีการเรียกใช้งานเพื่อหา nearest point ของข้อมูลจุดทดสอบจำนวน 1000 จุด ในจุดอ้างอิงทั้งหมด 1,000,000 จุด ใน 2 วิธีการคือ การสร้างลูปเพื่อป้อนข้อมูลทดสอบทีละจุด จนครบ 1000 จุด กับการสร้าง list ของจุดทดสอบ แล้วป้อนค่าดังกล่าวให้ library ทำงาน แล้วตอบคำถามว่าความเร็วในการทำงานของโปรแกรมส่วนที่เป็นตัวหนา เขียนโปรแกรมแบบไหนทำงานได้เร็วกว่า และเร็วกว่ากันกี่เท่า

| โปรแกรมที่ 1 | โปรแกรมที่ 2 |
|---|---|
| import numpy as np<br>from sklearn.neighbors import<br>NearestNeighbors<br>import random<br>import time<br><br>r = lambda: random.randint(0,10000) | import numpy as np<br>from sklearn.neighbors import<br>NearestNeighbors<br>import random<br>import time<br><br>r = lambda: random.randint(0,10000) |

| | |
|---|---|
| samples = [[r(),r(),r()] for i in range(1000000)]<br>testdat = [[r(),r(),r()] for i in range(1000)]<br>neigh = NearestNeighbors(n_neighbors=1)<br>neigh.fit(samples)<br><br>**detect1 = neigh.kneighbors(testdat)** | samples = [[r(),r(),r()] for i in range(1000000)]<br>testdat = [[r(),r(),r()] for i in range(1000)]<br>neigh = NearestNeighbors(n_neighbors=1)<br>neigh.fit(samples)<br><br>**detect2=[]**<br>**for i in testdat:**<br>**detect2 += neigh.kneighbors([i])** |

แล้วให้นักศึกษาลองวิเคราะห์ว่า การทำงานของทั้ง 2 โปรแกรมนั้นได้ผลลัพธ์ที่เหมือนกัน แต่เหตุใดจึงใช้เวลาในการทำงานที่แตกต่างกันมาก

**ตอนที่ 2 :** จงหา Big-O ของโปรแกรมต่อไปนี้

โปรแกรมที่ 1 :

```
n = input("input number : ")
n = int(n)
for i in range(1,n,2):
  print(i)
```

โปรแกรมที่ 2 :

```
n = input("input number : ")
n = int(n)
for x in range(1,n):
  for y in range(n - x):
    print (" ",end="")
  for y in range(1,x + 1):
    print(y,end="")
  for y in range(2,x + 1):
    print(x - y + 1,end="")
  print()
```

ชื่อ – นามสกุล .....รหัสนักศึกษา.....

โปรแกรมที่ 3 :

```
n = input("input number : ")
n = int(n)
i=1
while i<n:
  print(i)
  i=2*i
```

**ตอนที่ 3 :** เขียนโปรแกรมค้นหาข้อมูลให้มี Big-O ตามที่กำหนด

1. กำหนดให้ตัวแปร data เป็นค่า list ของค่าสุ่ม

```
import random
rnddat = [random.randint(1,1000) for i in range(0,1000000)]
```

แล้วให้นักศึกษาเขียนโปรแกรมเรียงตัวเลขใน rnddat จากน้อยไปมากโดยไม่ใช้คำสั่ง sort() โดยให้การทำงานของโปรแกรมนี้มีค่า Big-O เป็น O(n^2)

2. กำหนดให้ตัวแปร data มีค่าเป็น list ของตัวเลข 1-1,000,000

```
import random
dat = list(range(1,1000001))
```

แล้วให้นักศึกษาเขียนโปรแกรมรับ input เป็นตัวเลข 1 ตัวแล้วทำการค้นหาว่ามีตัวเลขดังกล่าวอยู่ที่ตำแหน่งใด โดยให้การทำงานของโปรแกรมนี้มีค่า Big-O เป็น O(log n)