มหาวิทยาลัยเอเชียอาคเนย์

ระดับ ☑ ปริญญาตรี ☐ ปริญญาโท ☐ ปริญญาเอก คณะบริหารธุรกิจ
หลักสูตร/สาขาวิชา สาขาวิชาสารสนเทศเพื่อธุรกิจดิจิทัล ชั้นปีที่ 1
ข้อสอบ ☐ กลางภาค ☑ ปลายภาค ประจำภาคการศึกษาที่ 2 ปีการศึกษา 2568 ภาคปกติ
รหัสวิชา 725103 ชื่อวิชา โครงสร้างข้อมูลและอัลกอริทึม จำนวน 3 หน่วยกิต คะแนนเต็ม 30 คะแนน
สอบวันที่ 16 พฤษภาคม 2569 เวลา 9:00-12:00 รวม 3 ชั่วโมง (ข้อสอบมีทั้งหมด 10 หน้า)

คำสั่ง

1. ให้นักศึกษาทำข้อสอบลงใน ☐ สมุดคำตอบ ☑ ข้อสอบ ☐ กระดาษคำตอบ
2. ☐ อนุญาตเปิดเอกสารระหว่างการสอบ ☑ ไม่อนุญาต ให้นำเอกสารทุกชนิดเข้าห้องสอบ
3. ข้อสอบทั้งหมดมี 1 ตอน 30 คะแนน
4. ☐ อนุญาต ☑ ไม่อนุญาต ให้ใช้เครื่องคำนวณทุกชนิด
5. ☐ อนุญาต ☑ ไม่อนุญาต ใช้คอมพิวเตอร์ โทรศัพท์มือถือ หรืออุปกรณ์สื่อสารทุกชนิด
6. ตอบเป็นภาษาไทยได้เต็มที่ แต่คำศัพท์เทคนิค ชื่อ algorithm, pseudocode และ asymptotic notation ให้คงเป็นภาษาอังกฤษ
7. ข้อสอบนี้ไม่ต้องเขียนโปรแกรมเต็มรูปแบบ แต่ต้องแสดงการคิด การ trace และเหตุผลให้ชัดเจน

หมายเหตุ:

- ข้อที่ให้เขียน pseudocode ไม่จำเป็นต้องตรงตาม syntax ของภาษาโปรแกรมใดภาษาหนึ่ง ขอเพียงลำดับขั้นตอนชัดเจน
- ในข้อที่เป็นการ trace code หรือ algorithm หากนักศึกษาไม่แสดงขั้นตอน อาจไม่ได้รับคะแนนเต็มแม้คำตอบสุดท้ายถูกต้อง

. . . . . .

ผู้ออกข้อสอบ
อาจารย์ปพนธีร์ แย้มโชติ
เบอร์โทร 087-049-2475

ชื่อ-สกุล . . . . . รหัสนักศึกษา . . . . . .

1. **OOP กับ data structure แบบประหลาด (5 คะแนน)**

พิจารณา pseudocode ต่อไปนี้

```
class DataBag:
    def __init__(self, name):
        self.name = name
        self.items = []

    def add(self, x):
        self.items.append(x)

    def remove(self):
        return self.items.pop()

    def first(self):
        return self.items[0]

    class MirrorBag(DataBag):
        def add(self, x):
            self.items.insert(0, x)

        def first(self):
            return self.items[-1]

if __name__ == "__main__":
    a = DataBag("A")
    b = MirrorBag("B")

    for v in [1, 2, 3]:
        a.add(v)
        b.add(v)

    r1 = a.remove()
    r2 = b.remove()
    f1 = a.first()
    f2 = b.first()
```

(a) (2 คะแนน) ระบุว่าในโค้ดนี้ class, object, attribute และ method ที่ถูก override คืออะไรบ้าง

(b) (1 คะแนน) หลังจบคำสั่ง for ค่าใน a.items และ b.items คืออะไร

(c) (1 คะแนน) ค่าของ r1, r2, f1, f2 คืออะไร

(d) (1 คะแนน) การนำข้อมูลออกจาก b มีพฤติกรรมเหมือน stack หรือ queue มากกว่า เพราะเหตุใด

ชื่อ-สกุล . . . . . รหัสนักศึกษา . . . . . .

---

2. **Recursion (6 คะแนน)**

2.1 อ่านโค้ด recursion และ trace การทำงาน (3 คะแนน)

พิจารณา pseudocode ต่อไปนี้

```
def mystery(n):
    if n <= 1:
        return 1
    if n % 2 == 0:
        return mystery(n / 2) + n
    return mystery(n − 1) + 1
```

(a) (0.5 คะแนน) base case คืออะไร

(b) (2 คะแนน) จง trace การทำงานของ mystery(6) โดยแสดงลำดับการเรียกฟังก์ชันและค่าที่ return ย้อนกลับ

(c) (0.5 คะแนน) ค่าผลลัพธ์สุดท้ายของ mystery(6) คืออะไร

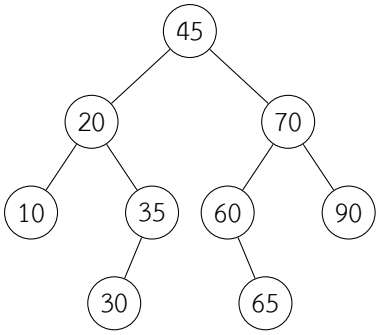
2.2 เขียน pseudocode เอง (3 คะแนน)

จงเขียน **recursive pseudocode** ของฟังก์ชัน height(node) เพื่อหาความสูงของ binary tree โดยกำหนดว่า

- ถ้า tree ว่าง ให้ height = -1
- ถ้าเป็น leaf node ให้ height = 0

3. **Binary Tree และ Binary Search Tree (5 คะแนน)**

กำหนด BST ดังรูป



(a) (1 คะแนน) จงเขียนลำดับการท่องต้นไม้แบบ Inorder

(b) (1 คะแนน) หากค้นหาค่า 65 ใน BST นี้ จะผ่าน node ใดบ้างตามลำดับ

(c) (2 คะแนน) หากเพิ่มค่า 25 ลงใน BST นี้ จงวาดต้นไม้ใหม่หลังการเพิ่มข้อมูล

(d) (1 คะแนน) หากค้นหาค่า 50 จะมีการเปรียบเทียบกับ node ใดบ้างก่อนสรุปว่าไม่พบ

ชื่อ-สกุล . . . . . รหัสนักศึกษา . . . . . .

---

4. **Graph Data Structure (4 คะแนน)**

กำหนดกราฟแบบไม่กำกับทิศทาง (undirected graph) ในรูป adjacency list ดังนี้

```
A: B, C
B: A, D, E
C: A, F
D: B, E
E: B, D, F, G
F: C, E
G: E
```

ให้ใช้กติกาว่า ถ้ามีเพื่อนบ้านหลายตัว ให้พิจารณาตามลำดับตัวอักษร

(a) (2 คะแนน) จงเขียนลำดับการท่องกราฟแบบ BFS โดยเริ่มที่ A

(b) (1 คะแนน) จงเขียนลำดับการท่องกราฟแบบ DFS โดยเริ่มที่ A

(c) (1 คะแนน) เส้นทางสั้นที่สุดจาก A ไป G คืออะไร

5. **Sorting Algorithm (10 คะแนน)**

5.1 Sorting พื้นฐาน (4 คะแนน)

พิจารณา pseudocode ต่อไปนี้

```
arr = [8, 6, 3, 1]

for i in range(0, len(arr) – 1):
    minIndex = i
    for j in range(i + 1, len(arr)):
        if arr[j] < arr[minIndex]:
            minIndex = j
        swap(arr[i], a1rr[minIndex])
```

(a) (1 คะแนน) pseudocode นี้คืออัลกอริทึมการเรียงลำดับแบบใด

(b) (1 คะแนน) หลังจบรอบนอก (outer loop) รอบที่ 1 ค่าใน arr จะเป็นอะไร

(c) (1 คะแนน) หลังจบรอบนอก (outer loop) รอบที่ 2 ค่าใน arr จะเป็นอะไร

(d) (1 คะแนน) Worst-case time complexity คืออะไร

5.2 ข้อพิเศษ: Sorting ที่ไม่ได้สอนในห้อง (6 คะแนน)

พิจารณา pseudocode ต่อไปนี้

```
arr = [5, 3, 4, 1]
i = 1

while i < len(arr):
    if i == 0 or arr[i − 1] <= arr[i]:
        i = i + 1
    else:
        swap(arr[i], arr[i − 1])
        i = i − 1
```

(a) (3 คะแนน) จง trace การทำงานโดยเขียนค่า arr **หลังการ swap แต่ละครั้ง** จนกระทั่งเรียงเสร็จ

ลำดับการ swap	ค่าใน arr หลังการ swap
1	
2	
3	
4	
5	
6	

(b) (1 คะแนน) ค่าใน arr เมื่อทำงานเสร็จทั้งหมดคืออะไร

(c) (2 คะแนน) จงวิเคราะห์ Worst-case time complexity ของ algorithm นี้ในรูป Big-O พร้อมอธิบายสั้น ๆ ว่าทำไม

ข้อโบนัส (+4 คะแนน) (ไม่ได้สอนในห้อง แต่อ่านทำความเข้าใจเองได้ไม่ยาก)

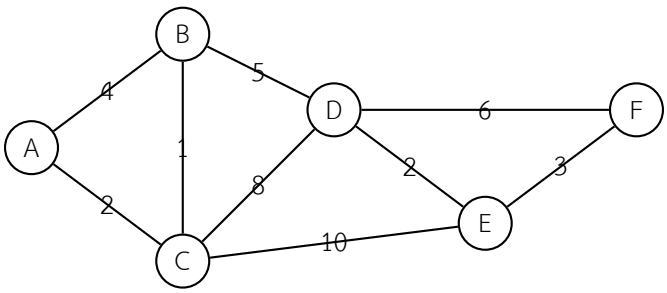
กำหนด *pseudocode* ของ **Dijkstra's Algorithm** ดังนี้ (ให้พิจารณาจาก pseudocode นี้ในการตอบคำถาม)

```
function Dijkstra(G, source):
    for each vertex v in G:
        dist[v] = ∞
        prev[v] = null
        visited[v] = false
    dist[source] = 0

    repeat |V| times:
        u = unvisited vertex with minimum dist
        visited[u] = true
        for each neighbor v of u:
            if not visited[v] and dist[u] + w(u,v) < dist[v]:
                dist[v] = dist[u] + w(u,v)
                prev[v] = u

    return dist, prev
```

กำหนดให้กราฟต่อไปนี้เป็นกราฟแบบไม่มีทิศทาง (undirected graph)



จงใช้ Dijkstra's Algorithm โดยเริ่มต้นที่จุดยอด **A** แล้วตอบคำถามต่อไปนี้

1. จง trace algorithm ทีละรอบ โดยกรอกค่าของ dist หลังจากเลือก vertex ในแต่ละรอบเสร็จแล้ว ลงในตารางต่อไปนี้ (3 คะแนน)

รอบ	vertex ที่เลือก	dist[A]	dist[B]	dist[C]	dist[D]	dist[E]	dist[F]
เริ่มต้น	--	0	∞	∞	∞	∞	∞
1							
2							
3							
4							
5							
6							

2. เส้นทางที่สั้นที่สุดจาก **A** ไป **F** คือเส้นทางใด และมีระยะทางรวมเท่าใด (1 คะแนน)