

ข้อสอบครั้งที่ ๒

ภาคเรียนที่ ๒ ปีการศึกษา ๒๕๕๙

วิชา การวิเคราะห์และการออกแบบขั้นตอนวิธี รหัส ๓๐๕๑๗๑

วันอาทิตย์ที่ ๑๙ มีนาคม พ.ศ. ๒๕๖๐ เวลา ๑๓.๐๐ – ๑๖.๐๐ น. ห้องเรียน EE-606

ชื่อ-สกุลของนิสิต..... รหัส.....

คำชี้แจงสำหรับการสอบ (นิสิตจะต้องอ่านให้เข้าใจก่อนทำข้อสอบ และเซ็นชื่อกำกับด้านล่าง มิฉะนั้น จะไม่ได้รับการพิจารณาตรวจข้อสอบ)

1. นิสิตต้องนั่งตามผังที่นั่งสอบที่ติดไว้หน้าห้องสอบ
2. นิสิตที่เข้าสอบจะต้องนำบัตรนิสิตติดตัวมาด้วยทุกครั้ง เพื่อให้ผู้คุมสอบตรวจสอบได้ และนิสิตจะต้องเซ็นชื่อลงในใบเซ็นชื่อด้วยปากกา เพื่อยืนยันการเข้าสอบ
3. ห้ามนิสิตเข้าสอบช้ากว่าเวลา 15 นาที และไม่ให้ออกจากห้องสอบก่อน 30 นาทีหลังเริ่มสอบ เว้นเสียแต่จะได้รับอนุญาตเป็นลายลักษณ์อักษรจากผู้คุมสอบหรือคณบดี
4. ข้อสอบมี 3 ข้อ 3 หน้า (รวมใบปะหน้า)
5. ในการสอบนี้อนุญาตให้นิสิตนำเอกสาร เครื่องเขียน และเครื่องคำนวณที่ไม่ได้ต่อกับสัญญาณสื่อสารใดๆ เข้าห้องสอบได้
6. ไม่อนุญาตให้ใช้/เปิดโทรศัพท์มือถือระหว่างทำข้อสอบ ทั้งนี้รวมถึงไม่อนุญาตให้นิสิตใช้โทรศัพท์มือถือในการคำนวณแทนเครื่องคิดเลข มิฉะนั้นจะถือว่านิสิตมีเจตนาทุจริต
7. ห้ามเยี่ยมของกัน และห้ามนิสิตพูดคุยกันระหว่างการทำข้อสอบ มิฉะนั้น จะถือว่าส่อเจตนาทุจริต
8. เมื่อปรากฏว่ามีการทุจริตในการสอบ ผู้ควบคุมการสอบจะรวบรวมพยานหลักฐาน และบันทึกลักษณะความผิดไว้ในกระดาษคำตอบ พร้อมกับลงลายมือชื่อรับรอง แล้วให้รับรายงานคณบดี เพื่อนำเสนอมหาวิทยาลัยพิจารณาดำเนินการตามข้อบังคับของมหาวิทยาลัยว่าด้วยการสอบของนิสิตต่อไป

ข้าพเจ้าได้อ่านคำชี้แจงในการทำข้อสอบและทำความเข้าใจเป็นอย่างดีแล้ว และพร้อมปฏิบัติตาม

ลงชื่อ

นิสิตผู้เข้าสอบ

1. (StringMatch) กำหนดให้ text คือสายอักขระ (string) ตั้งต้นมีความยาว n และแบบรูปที่ต้องการค้นหา (search pattern) คือสายอักขระ (string) ที่มีความยาว m ($m \leq n$) จงออกแบบขั้นตอนวิธีในจับคู่สายอักขระย่อย (substring) ของ text ที่เท่ากับ pattern การค้นหาตำแหน่งของ pattern ใน text ถ้า pattern ไม่ปรากฏใน text เลย ให้ผลลัพธ์เป็น -1 ตัวอย่างเช่น

StringMatch("somchai jaidee", "jai") == 8

StringMatch("somchai jaidee", "somsak") == -1

StringMatch("somchai jaidee", "Somchai") == -1

- 1.1. จงกำหนดเงื่อนไขหลัง (postcondition) ของฟังก์ชัน StringMatch
- 1.2. จงออกแบบฟังก์ชัน verifyMatch เพื่อตรวจสอบว่าผลลัพธ์ที่ได้จาก StringMatch ถูกต้องหรือไม่
- 1.3. จงออกแบบ StringMatch ด้วยแนวคิด Brute-force แต่ถ้าคุณออกแบบด้วยแนวคิดอื่นให้ระบุด้วยว่าใช้แนวคิดใด
- 1.4. จงวิเคราะห์ความซับซ้อนของขั้นตอนวิธีที่ออกแบบมา
2. สมมติว่ามีแก้ว $2n$ ใบวางอยู่บนโต๊ะ แก้ว n ใบมีน้ำอยู่ในขณะที่แก้วอีก n ใบว่างเปล่า จงออกแบบขั้นตอนวิธี (arrangeCup) ในการที่จะเทน้ำจากแก้วที่มีน้ำอยู่ไปยังแก้วว่างเปล่าแล้วผลลัพธ์เป็นแก้วที่มีน้ำอยู่ด้านซ้ายในขณะที่แก้วเปล่าอยู่ด้านขวา ตัวอย่างเช่นถ้ามีแก้ว 10 ใบ ดังภาพ



เมื่อเทน้ำเสร็จแล้วจะผลลัพธ์ดังภาพ



- 2.1. จงกำหนดเงื่อนไขหลัง (postcondition) ของฟังก์ชัน arrangeCup
- 2.2. จงออกแบบฟังก์ชัน verifyArrange เพื่อตรวจสอบว่าผลลัพธ์ที่ได้จาก arrangeCup ถูกต้องหรือไม่
- 2.3. จงออกแบบ arrangeCup ด้วยแนวคิด Decrease and Conquer แต่ถ้าคุณออกแบบด้วยแนวคิดอื่นให้ระบุด้วยว่าใช้แนวคิดใด
- 2.4. จงวิเคราะห์ความซับซ้อนของขั้นตอนวิธีที่ออกแบบมา
3. ถ้า Array ของ int ขนาด n มีค่าใน array ตัวหนึ่งปรากฏเกินกว่าครึ่ง เราจะเรียก Array นี้ว่ามี majority element จงออกแบบฟังก์ชัน hasMajoriyElement สำหรับพิจารณาว่า Array ใดๆ มี majority element หรือไม่
- 3.1. จงกำหนดเงื่อนไขหลัง (postcondition) ของฟังก์ชัน hasMajoriyElement

3.2. จงเขียนฟังก์ชัน verifyME เพื่อตรวจสอบว่าผลลัพธ์ที่ได้จาก hasMajoriyElement ถูกต้องหรือไม่

3.3. จงออกแบบ hasMajoriyElement ด้วยแนวคิด Divide and Conquer แต่ถ้าคุณออกแบบด้วยแนวคิดอื่นให้ระบุว่าใช้แนวคิดใด

3.4. จงวิเคราะห์ความซับซ้อนของขั้นตอนวิธีที่ออกแบบมา

Hint: คุณอาจจะทำได้ทั้ง

* แบบ Top-down (ในแบบของ Quick Sort) โดยแบ่งเป็น Array เป็น 2 ส่วนแล้ว แบ่งทีละส่วนย่อยลงไปเรื่อยๆ

* แบบ Bottom-up (ในแบบของ Merge Sort) โดยเปรียบเทียบทีละ 2 ค่า ถ้าเหมือนกันก็เก็บไว้ 1 ค่า ถ้าต่างก็โยนทิ้งทั้ง 2 ค่า