String Processing

Overview

- String processing นั้นเป็นสิ่งที่พบมากสำหรับงานทางด้าน bioinformatics เมื่อ string (สามารถมองได้เป็นสาย DNA) ที่นักวิจัยมองนั้นเกี่ยวกับอัลกอริทึมและโครงสร้างข้อมูลที่เก็บข้อความที่ยาวที่ประมวลผลได้อย่างมีประสิทธิภาพ
- หลายๆ ปัญหานั้นถูกนำเอามาเป็นปัญหาแข่งขัน
- เริ่มต้นจะมาดู basic string processing หลังจากนั้นจะมาดูพวก Ad Hoc string problem

Basic string processing

- ในหัวข้อนี้จะให้ค่อยๆ ทำโจทย์เล็กๆ ไปทีละข้อ ให้ลองพยายามให้ code สั้นและมีประสิทธิภาพมากที่สุด จากนั้นจะมาเทียบกับตัวอย่างในตอนท้าย
- **ข้อแรก** กำหนด text file มาให้ที่ข้างในมีเพียงอักขระ [A–Za–z], ตัวเลข [0–9], space และ period('.') เท่านั้น จงเขียนโปรแกรมที่อ่าน text file นี้ทีละบรรทัด จนกระทั่งพบบรรทัดที่ขึ้นต้นบรรทัดด้วย จุด 7 จุด ('.......') ให้เชื่อม (Concatenate) แต่ละบรรทัดเป็นข้อความ T เมื่อ 2 บรรทัดถูกรวมกันให้เพิ่ม 1 space ระหว่างพวกมัน เพื่อให้อักขระสุดท้ายของข้อความแรกห่างจากอักขระแรกของข้อความที่สอง ในแต่ละบรรทัดมีได้ไม่เกิน 30 อักขระและไม่เกิน 10 บรรทัด แต่ละบรรทัดจบด้วย newline character

- **ตัวอย่างไฟล์**

I love CS204355 Competitive

Programming. i also love

AlGoRiThM

.....you must stop after reading this line as it starts with 7 dots

after the first input block, there will be one looooooooooong line...

ตอบคำถามเหล่านี้

- จะเก็บ string อย่างไร
- อ่านข้อความ input ทีละบรรทัดอย่างไร
- เชื่อมต่อกันสองข้อความทำอย่างไร
- จะตรวจสอบว่าถ้าบรรทัดนั้นขึ้นต้นด้วย '.....' แล้วหยุดอ่านทำอย่างไร

- **<u>ข้อสอง</u>** สมมติว่าเราต้องการทำการวิเคราะห์อักขระในข้อความ T และต้องการเปลี่ยนแต่ละอักขระใน T ให้เป็นอักขระตัวเล็ก

- การวิเคราะห์ที่ต้องการคือ มีตัวเลขกี่ตัว สระ[aeiouAEIOU]กี่ตัว consornant(อักขระอื่นที่ไม่ใช้สระ)กี่ตัว ใน T สามารถตรวจสอบสิ่งเหล่านี้ได้ใน O(n) เมื่อ n เป็นความยาวของ string หรือไม่

- **<u>ข้อสาม</u>** สมมติว่าเรามีข้อความยาวๆ T เราต้องการตรวจสอบว่าอีกข้อความ P สามารถถูกพบได้ใน T หรือไม่ ให้ระบุทุก index ที่ P ปรากฏใน T หรือแจ้ง –1 ถ้า P ไม่พบใน T

- ตัวอย่างเช่น ถ้า T='I love CS204355 Competitive Programming. I also love AlGoRiThM' และ P='I' แล้ว output เป็น {0} แต่ถ้า P='love' output เป็น {2, 48} และถ้า P='book' แล้ว output เป็น {NOT FOUND}

ตอบคำถามเหล่านี้

- จะหาการเกิดขึ้นครั้งแรกของ substring ใน string ได้อย่างไร เราต้อง implement string matching algorithm (Knuth–Morris–Pratt algorithm) ไหมหรือใช้แค่ library function ได้

- แล้วจะหาการเกิดขึ้นครั้งต่อ ๆ ไปของ substring ใน string

- **<u>ข้อสี่</u>** เราต้องการแยกข้อความยาวๆ T ออกเป็น tokens(substring) และเก็บไว้ใน array of string ที่เรียกว่า tokens ในข้อนี้ delimiter ของ token เหล่านี้คือ space หรือ period (ดังนั้นจะแตกประโยคเป็นคำ)
- ตัวอย่างเช่น ถ้าเราแตก string (tokenize) T ในรูปแบบตัวเล็กเราจะได้ tokens ={'i', 'love', 'cs204355', 'competitive', 'programming', 'i', 'also', 'love', 'algorithm'} จากนั้นเราจะ sort array นี้ตามลำดับอักษร (lexicographically) และหา lexicographically smallest string นั่นคือเรา sort tokerns ={'algorithm', 'also', 'competitive', 'cs204355', 'i', 'i', 'love', 'love', 'programming'} ดังนั้น lexicographically smallest string คือ algorithm

ตอบคำถามเหล่านี้

- แตก string อย่างไร
- เก็บ token ใน array string อย่างไร
- เรียง array of string ตามลำดับอักขระ Lexicography ได้อย่างไร

- **<u>ข้อห้า</u>** ระบุว่าคำใดพบมากที่สุดใน T ในการที่จะตอบการสอบถามนี้นั้น เราต้องการนับความถี่ของแต่ละคำ สำหรับ T output จะเป็นได้ทั้ง 'I' และ 'love' เพราะว่าพบทั้งสองครั้งเท่ากัน

- โครงสร้างข้อมูลแบบใดที่ควรถูกใช้ในงานนี้

- **ข้อหก** จาก text file ที่กำหนดมาให้มีอีกบรรทัดหนึ่งหลังจากบรรทัดที่เริ่มต้นด้วย '.....' แต่ความยาวของบรรทัดสุดท้ายนี้ไม่ได้ระบุข้อจำกัดไว้ งานของคุณคือนับจำนวนอักขระที่มีอยู่ในบรรทัดสุดท้าย
- เราจะนับข้อความถ้าความยาวของมันไม่รู้มากก่อนได้อย่างไร