**Page Replacement Algorithms**

**นายกฤษณะ มะนุภา**

**รหัส 600612147 Section 801**

**รายงานกระบวนวิชา Operating System**

**ภาควิชาวิศวกรรมคอมพิวเตอร์**

**คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่**

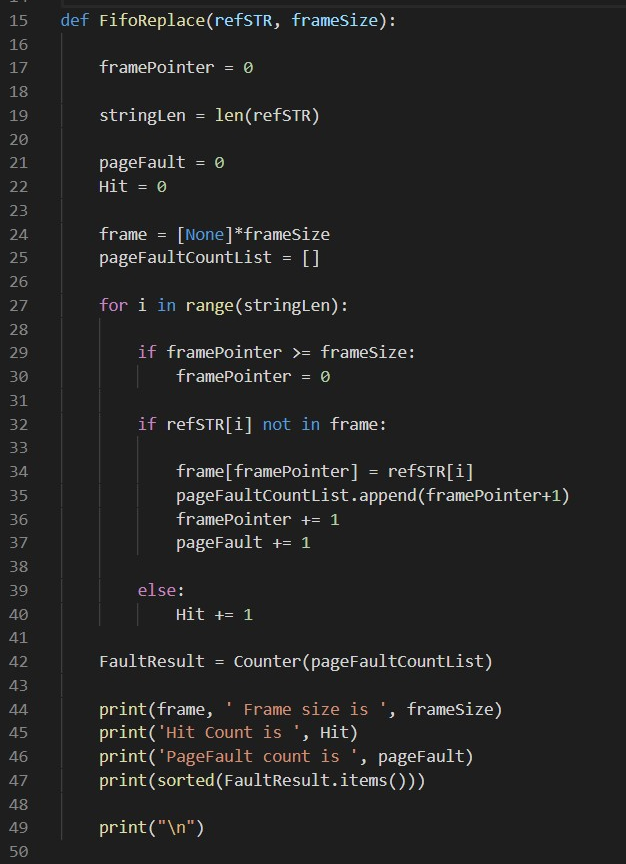
**ภาคการศึกษาที่ 2/2562**

**First In First Out (FIFO)**

**แนวคิด**

หลักการของ Algorithm นี้คือการวนใช้ Index ใน Frame ตามลำดับ จะเริ่มใหม่ก็ต่อเมื่อวนไปจนถึง Index ท้ายสุด จึงได้มีการกำหนดตัวแปร framePointer มาทำหน้าที่ระบุ Index ที่จะให้ Reference strings มาเพิ่มหรือแทนที่ ณ Index นั้น ๆ ในส่วนต่อไปจำเป็นที่จะต้องมีตัวเก็บค่านับจำนวนเพื่อแสดง Hit หรือ PageFault ได้ในแต่ละ Index คือตัวแปร pageFaultCountList โดยเก็บเป็น List และนำมาใช้ฟังก์ชัน Counter เพื่อนับรวม

**CODE**

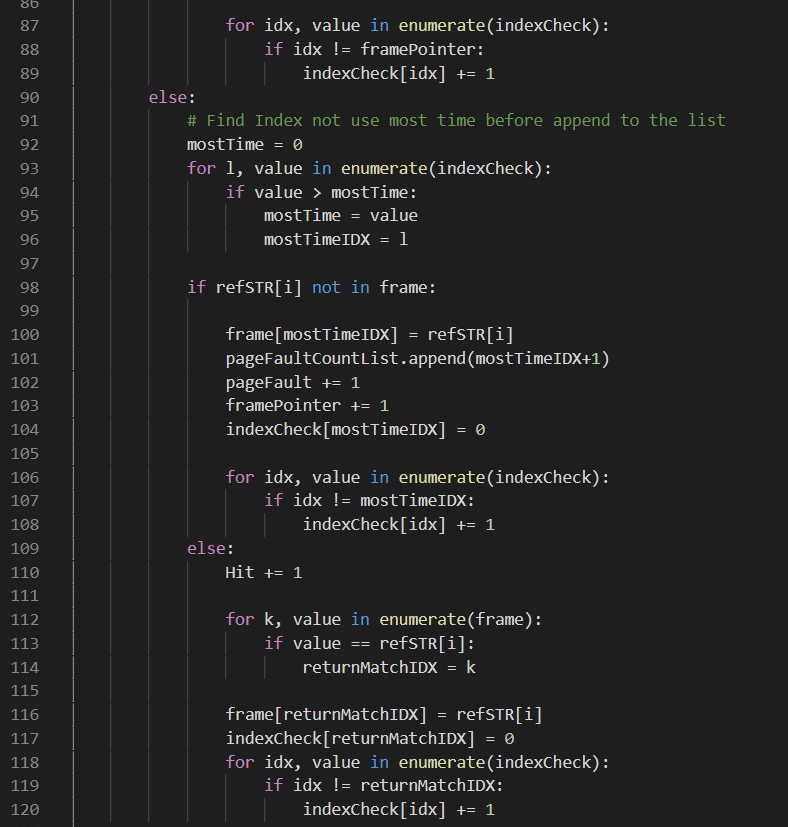
****

**Optimal**

**แนวคิด**

เนื่องจาก Optimal Algorithm นี้จะเกี่ยวเนื่องกับการจดจำ Index ภายใน Frame จึงได้สร้างตัวแปร indexCheck ตามจำนวน frameSize ขึ้นมา ภายใน Function ของ Algorithm นี้ภายในจะแบ่งเป็น 2 ส่วนใหญ่ ๆ ในส่วนแรกหมายถึงหาก Frame ยังไม่เต็ม ในส่วนที่สองหมายถึง เมื่อ Frame เต็มแล้ว ซึ่งในสองส่วนนี้ จะมีเงื่อนไขย่อยอีก คือ ยังไม่เคยมี String นั้น ๆ ใน Frame หรือมี String นั้น ๆ อยู่ใน Frame อยู่แล้ว ในการทำงาน indexCheck จะเป็น List ซึ่งเก็บตัวเลขจำนวนเต็มอยู่ตรงตาม Index ของ Frame แต่ละรอบทุก ๆ ค่าใน indexCheck จะเพิ่มขึ้น 1 ยกเว้น ณ Index ที่มีการเพิ่มเข้ามาใน Frame ล่าสุด หรือมีการ Hit จะถูกกำหนดค่าใน indexCheck ให้เป็น 0 หาก Frame เต็ม จะทำการเปรียบเทียบจาก indexCheck จาก Index ไหนมากที่สุด หมายความว่า Index นั้น ไม่ได้ถูกใช้งานนานแล้ว ส่งผลให้จะถูกแทนที่ด้วย String ชุดที่เข้ามาใหม่

**CODE**

** **

**Least recently used (LRU)**

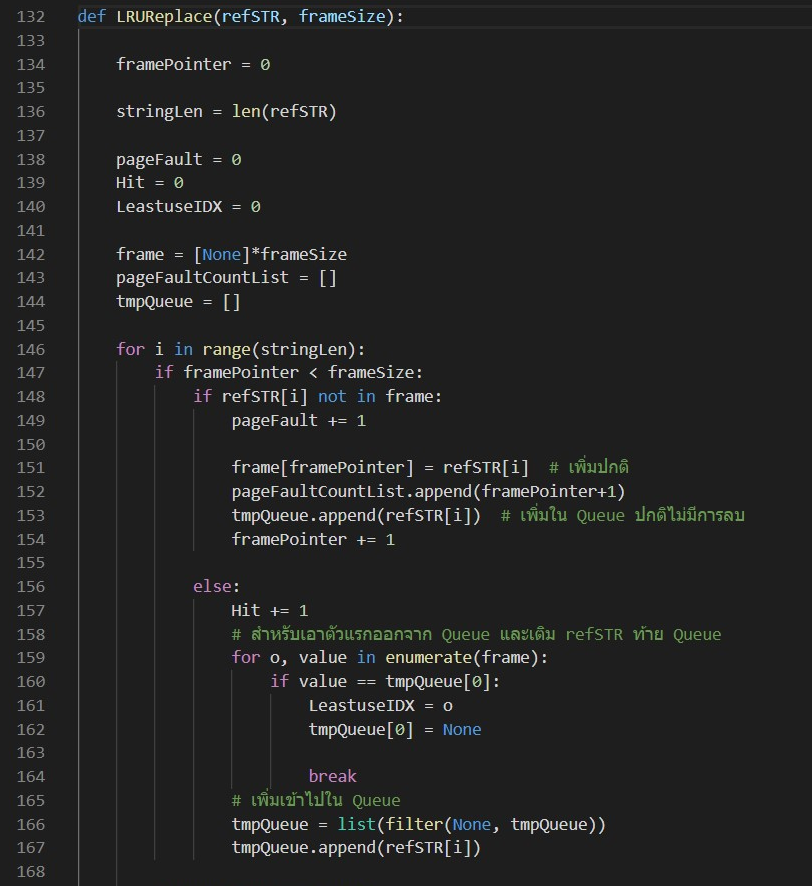
**แนวคิด**

เนื่องจาก LRU Algorithm เป็นการโฟกัสไปที่ Queue ของ String ใน Frame จึงได้กำหนดตัวแปร tmpQueue ขึ้นมาเป็น Type List สาเหตุที่ไม่ใช้ Type Queue โดยตรง เพราะ Queue ไม่สามารถนำค่าที่อยู่ระหว่าง Queue ออกมาได้ เพื่อสะดวกต่อการใช้งานจึงเลือก List เป็นเสมือน Queue แทน ฟังก์ชันนี้ภายใน Loop มีสองส่วนใหญ่ ๆ ส่วนแรกคือ เมื่อ Frame ยังไม่เต็ม ส่วนที่สอง เมื่อ Frame เต็มแล้ว และส่วนย่อยในสองส่วนหลัก ๆ นี้ ประกอบไปด้วย เงื่อนไขหากไม่มี String นั้น ๆ อยู่ใน Frame และ เงื่อนไขที่เมื่อมี String นั้น ๆ คงอยู่ใน Frame อยู่แล้ว

ในการทำงานเมื่อ Frame ยังไม่เต็ม และไม่เคยมี String อยู่ใน Frame จะส่งค่า String ไปเก็บไว้ทั้งใน Frame และ tmpQueue หาก String ไหนมีอยู่ใน Frame ไม่ว่าจะอยู่ Index ไหนก็ตาม จะทำการกำหนดให้ ค่าของ String นั้น ๆ เป็น None และใช้ฟังก์ชัน filter ช่วยในการลบ None ออกจากใน Queue ให้เรียงตามลำดับติดกันไม่ถูกแทรกด้วย None ( ขยับ Queue มาติดกัน ) และ ทำการเพิ่มค่า String นั้น ๆ มาต่อท้ายใหม่ เสมือนการ Reset สถานะจดจำว่า String ค่านี้พึ่งมีการใช้งานใน Frame ล่าสุด

ในการนำไปใช้กำหนดค่าใน Frame นั้นได้จากการนำ String ตัวแรกของ tmpQueue มาตรวจเงื่อนไขใน Frame ให้มี Value ตรงกัน เพื่อที่จะนำ Index ของ String Value ใน Frame นั้น ๆ ถูกแทนที่ด้วย String ใหม่ ณ ตำแหน่ง Index ตามเงื่อนไข

**CODE**

****

****

**การทดลอง**

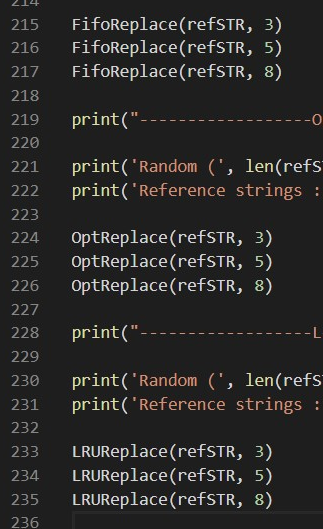
// ใส่ค่าในฟังก์ชัน

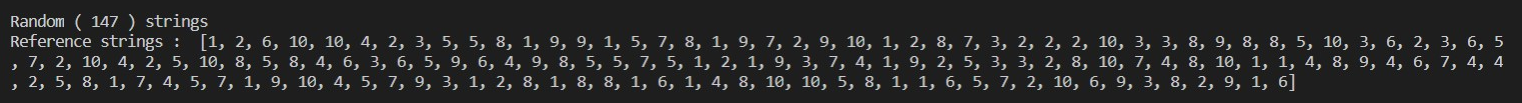
**สมมุติฐาน**

//เหตุผลคร่าวๆ ที่เล็งเห็นว่า ALG ไหนดี

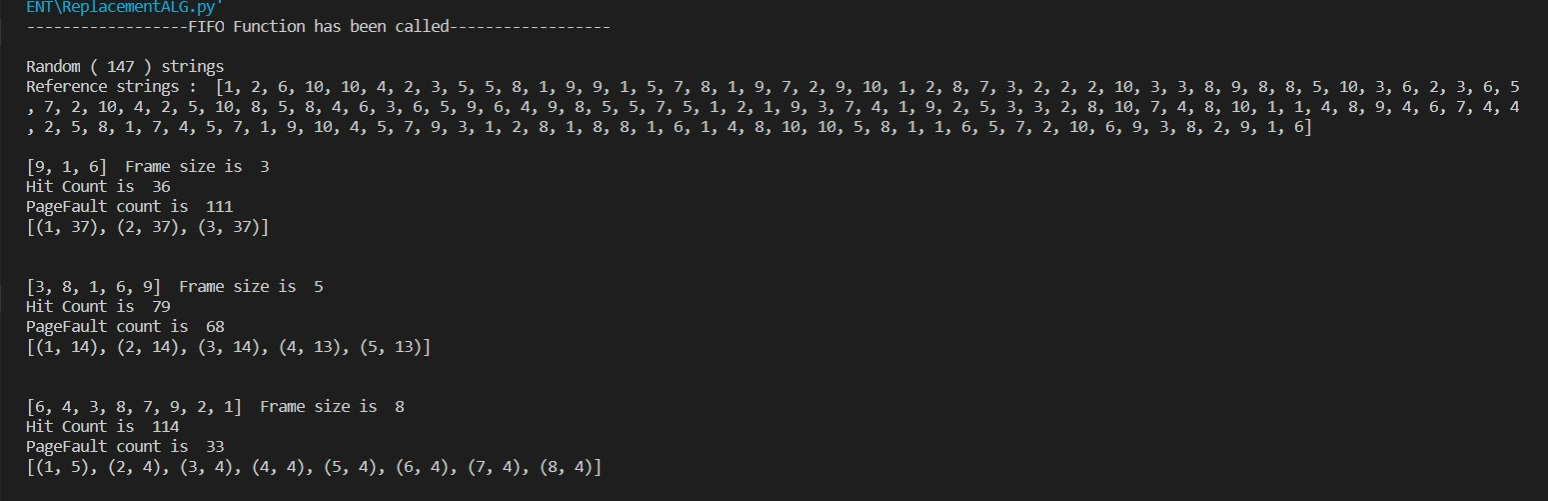
**Reference string**

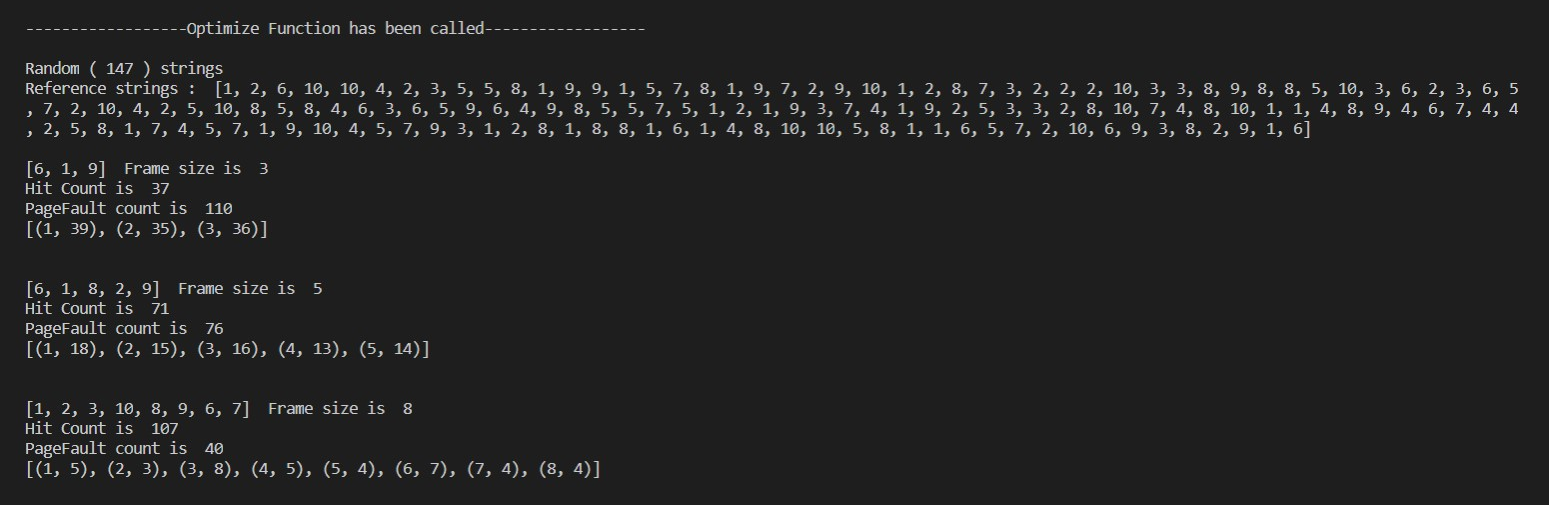
//Gen str

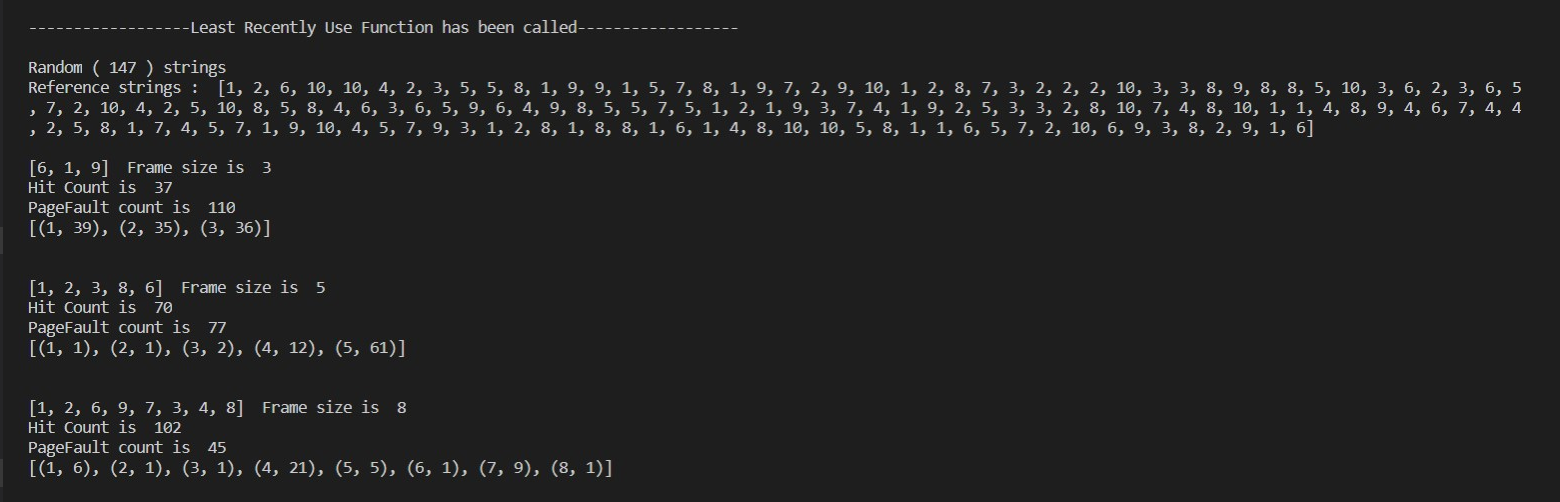




**ประเมินผลการทดลอง**

****

****

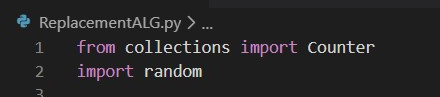
****

**สรุปผลการทดลอง**

//อธิบายจากการประเมินและสมมุติฐาน

**ภาคผนวก**

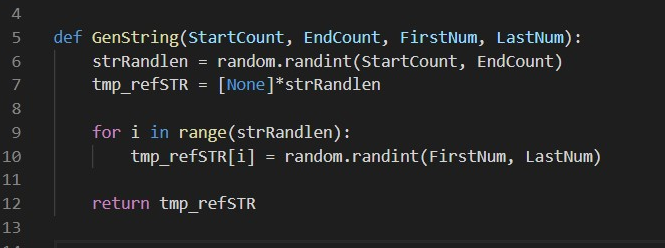
**Library**

****

Counter ใช้สำหรับการนับรวมแต่ละ Index ใน Frame ที่เกิด Page fault

Random ใช้สำหรับการ Generate ค่าต่าง ๆ เพื่อสร้าง Reference String

**Code Generating String**



**Tool**

Python 3.7.4

**GITHUB**

https://github.com/guylaxy31/Replacement3ALG