

Process Scheduling 2 (scheduling2)

[Time Limit : 1 sec , Mem Limit : 32 MB]

Problem :

ซีพียู (CPU) คือหน่วยประมวลผลกลางของเครื่องคอมพิวเตอร์ซึ่งต้องทำงานต่างๆมากมาย โดยงานต่างๆที่ซีพียูต้องทำจะเกิดจากสิ่งที่เรียกว่า โปรเซส (Process) ในขณะที่ขณะหนึ่งซีพียูจะสามารถทำงานได้เพียง 1 งานเท่านั้น ดังนั้นหากมีมากกว่า 1 โปรเซส ที่ต้องการเข้าไปทำงานในซีพียู ก็จะต้องมีอัลกอริทึมสำหรับการทำ Process Scheduling เพื่อเลือกลำดับว่าโปรเซสใดจะได้เข้าไปทำงานในซีพียูก่อน (หรือหลัง)

หนึ่งในอัลกอริทึมพื้นฐานคือ RR (Round Robin) โดยปกติแล้วจะมีแถวสำหรับให้โปรเซสต่อแถวเพื่อรอเข้าไปทำงานในซีพียู เรียกว่า Wait Queue และจะมีการกำหนดจำนวนเต็ม 1 ตัว เรียกว่า Time Slice โดยที่ช่วงเวลา t ใดๆ (t เป็นจำนวนเต็มที่มากกว่า 0) จะมีกฎในการจัดการโปรเซสต่างๆ ตามขั้นตอนดังนี้

1. หากมีโปรเซสที่เพิ่งเกิดขึ้นใหม่ ให้โปรเซสนั้นเข้ามาต่อท้ายแถวใน Wait Queue
2. หากมีโปรเซสที่กำลังทำงานอยู่ในซีพียู กรณีที่โปรเซสนั้นทำงานเสร็จแล้ว ก็ให้โปรเซสนั้นออกจากซีพียูไป
3. หากมีโปรเซสที่กำลังทำงานอยู่ในซีพียู และโปรเซสนั้นยังทำงานไม่เสร็จ และอยู่ในซีพียูมาเป็นระยะเวลาอย่างไม่เกินค่า Time Slice ก็ให้ซีพียูทำงานของโปรเซสนั้นๆต่อไป แต่หากอยู่ในซีพียูมาเป็นระยะเวลาเกินค่า Time Slice โปรเซสนั้นจะถูกบังคับให้ออกจากซีพียู และไปต่อท้ายแถวใน Wait Queue
4. หากไม่มีโปรเซสที่กำลังทำงานอยู่ในซีพียู ให้นำโปรเซสที่อยู่หัวแถวของ Wait Queue เข้ามาทำงานในซีพียู กรณีที่ Wait Queue ว่างก็ให้ซีพียูอยู่ในสถานะว่างต่อไป

ทั้งนี้สำหรับอัลกอริทึมใดๆย่อมมีการวัดประสิทธิภาพ ซึ่งสำหรับการทำ Process Scheduling นี้ก็จะมีวิธีวัดประสิทธิภาพจากการคำนวณ Average Waiting Time ซึ่งก็คือค่าเฉลี่ยของเวลาที่ทุกๆโปรเซสต้องรออยู่ใน Wait Queue จงเขียนโปรแกรมเพื่อรับลำดับเวลาการเกิดของโปรเซส และเวลาทั้งหมดที่คาดว่าโปรเซสนั้นๆจะใช้ในการทำงานในซีพียู และคำนวณ Average Waiting Time ออกมา

Input :

บรรทัดแรก ระบุจำนวนเต็ม T ($1 \leq T \leq 10$) แทนจำนวนชุดทดสอบย่อยทั้งหมด

สำหรับแต่ละชุดทดสอบย่อย :

บรรทัดแรก ระบุจำนวนเต็ม N, M ($1 \leq N \leq 5,000$; $1 \leq M \leq 50$) แทนจำนวนโปรเซสทั้งหมด และค่า Time Slice ตามลำดับ

อีก N บรรทัด โดยบรรทัดที่ i ($1 \leq i \leq N$) ระบุจำนวนเต็ม P_i, Q_i ($1 \leq P_i \leq 5,000,000$; $1 \leq Q_i \leq 1,000$) แทนเวลาเกิดและระยะเวลาที่จะใช้ในการทำงานทั้งหมดของโปรเซสที่ i ตามลำดับ โดยรับประกันว่า $P_i < P_{i+1}$ เสมอ

Output :

สำหรับแต่ละชุดทดสอบย่อย มีบรรทัดเดียว ให้แสดงค่า Average Waiting Time โดยมีทศนิยม 7 ตำแหน่ง

Example :

Sample Input	Sample Output
1 3 3 2 4 3 5 8 7	4.0000000
2 4 2 1 3 5 5 6 2 10 1 3 5 1 29 5 37 10 16	1.2500000 39.3333333

อธิบายตัวอย่างที่ 1

มี 1 ชุดทดสอบย่อย โดยชุดทดสอบย่อยนั้นมีทั้งหมด 3 โปรเซส และมี Time Slice = 3

ณ เวลา $t = 2$;

โปรเซสที่ 1 เกิดขึ้นมา (ซึ่งคาดว่าจะใช้เวลาในการทำงานทั้งหมดอีก 4) เข้ามาต่อท้ายใน Wait Queue (ตามกฎข้อ 1)

เนื่องจากซีพียูว่างงานอยู่ ดังนั้นจากดีงโปรเซสที่หัวแถวเข้ามาทำงาน (ตามกฎข้อ 4)

Waiting Time สะสม ($P1 = 0$)

CPU	Wait Queue				
P1 (Q = 4)					

ณ เวลา $t = 3$;

โปรเซสที่ 2 เกิดขึ้นมา (ซึ่งคาดว่าจะใช้เวลาในการทำงานทั้งหมดอีก 5) เข้ามาต่อท้ายใน Wait Queue (ตามกฎข้อ 1)

Waiting Time สะสม ($P1 = 0, P2 = 0$)

CPU	Wait Queue				
P1 (Q = 3)	P2 (Q = 5)				

ณ เวลา $t = 5$;

โปรเซสที่ 1 ใช้เวลาในซีพียูนานเท่ากับ 3 (ยังเหลือเวลาในการทำงานอีก 1) จึงออกจากซีพียูและไปต่อท้ายแถว (ตามกฎข้อ 3)

เนื่องจากซีพียูว่างงานอยู่ ดังนั้นจากดีงโปรเซสที่หัวแถวเข้ามาทำงาน (ตามกฎข้อ 4)

Waiting Time สะสม ($P1 = 0, P2 = 2$)

CPU	Wait Queue				
P2 (Q = 5)	P1 (Q = 1)				

ณ เวลา $t = 8$;

โปรเซสที่ 3 เกิดขึ้นมา (ซึ่งคาดว่าจะใช้เวลาในการทำงานทั้งหมดอีก 7) เข้ามาต่อท้ายใน Wait Queue (ตามกฎข้อ 1)

โปรเซสที่ 2 ใช้เวลาในซีพียูนานเท่ากับ 3 (ยังเหลือเวลาในการทำงานอีก 2) จึงออกจากซีพียูและไปต่อท้ายแถว (ตามกฎข้อ 3)

เนื่องจากซีพียูว่างงานอยู่ ดังนั้นจากดิ่งโปรเซสที่หัวแถวเข้ามาทำงาน (ตามกฎข้อ 4)

Waiting Time สะสม ($P1 = 3, P2 = 2, P3 = 0$)

CPU	Wait Queue				
P1 (Q = 1)	P3 (Q = 7)	P2 (Q = 2)			

ณ เวลา $t = 9$;

โปรเซสที่ 1 ทำงานเสร็จ จึงออกจากซีพียูไป (ตามกฎข้อ 2)

เนื่องจากซีพียูว่างงานอยู่ ดังนั้นจากดิ่งโปรเซสที่หัวแถวเข้ามาทำงาน (ตามกฎข้อ 4)

Waiting Time สะสม ($P1 = 3, P2 = 3, P3 = 1$)

CPU	Wait Queue				
P3 (Q = 7)	P2 (Q = 2)				

ณ เวลา $t = 12$;

โปรเซสที่ 3 ใช้เวลาในซีพียูนานเท่ากับ 3 (ยังเหลือเวลาในการทำงานอีก 4) จึงออกจากซีพียูและไปต่อท้ายแถว (ตามกฎข้อ 3)

เนื่องจากซีพียูว่างงานอยู่ ดังนั้นจากดิ่งโปรเซสที่หัวแถวเข้ามาทำงาน (ตามกฎข้อ 4)

Waiting Time สะสม ($P1 = 3, P2 = 6, P3 = 1$)

CPU	Wait Queue				
P2 (Q = 2)	P3 (Q = 4)				

ณ เวลา $t = 14$;

โปรเซสที่ 2 ทำงานเสร็จ จึงออกจากซีพียูไป (ตามกฎข้อ 2)

เนื่องจากซีพียูว่างงานอยู่ ดังนั้นจากดิ่งโปรเซสที่หัวแถวเข้ามาทำงาน (ตามกฎข้อ 4)

Waiting Time สะสม ($P1 = 3, P2 = 6, P3 = 3$)

CPU	Wait Queue				
P3 (Q = 4)					

ณ เวลา $t = 17$;

โปรเซสที่ 3 ใช้เวลาในซีพียูนานเท่ากับ 3 (ยังเหลือเวลาในการทำงานอีก 1) จึงออกจากซีพียูและไปต่อท้ายแถว (ตามกฎข้อ 3)

เนื่องจากซีพียูว่างงานอยู่ ดังนั้นจากดิ่งโปรเซสที่หัวแถวเข้ามาทำงาน (ตามกฎข้อ 4)

Waiting Time สะสม ($P1 = 3, P2 = 6, P3 = 3$)

CPU	Wait Queue				
P3 (Q = 1)					

ณ เวลา $t = 18$;

โปรเซสที่ 2 ทำงานเสร็จ จึงออกจากซีพียูไป (ตามกฎข้อ 2)

Waiting Time สะสม ($P1 = 3, P2 = 6, P3 = 3$)

CPU	Wait Queue				
Free					

ดังนั้น $\text{Average Waiting Time} = (3 + 6 + 3) / 3 = 4.0000000$