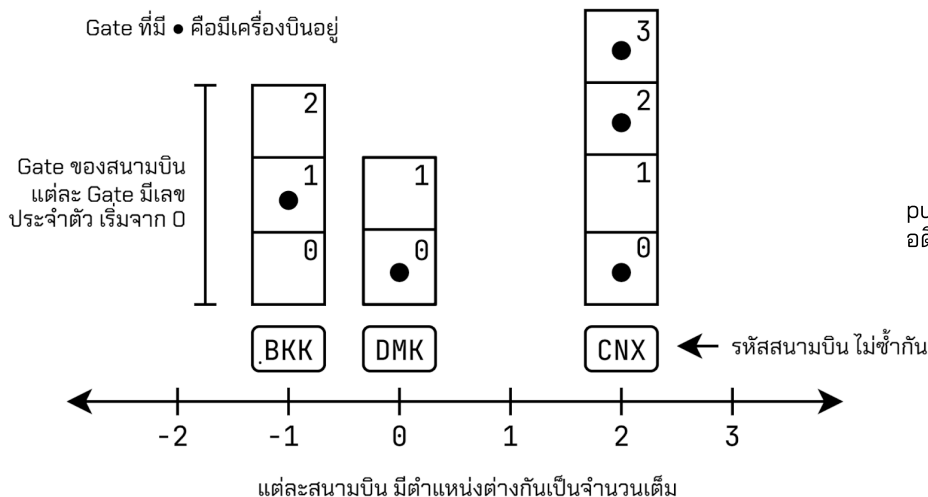


## Linear Airports: Merging Chaos

ณ ชมพูทวีปเส้นตรง มีผู้คนอยู่ทุกหนแห่งบนแผ่นดิน เจ้าแคว้นนามว่า **เอ็นหจิก** จึงออกนโยบายให้ทุกๆ หัวเมืองมีสนามบินเพื่อให้ประชาชนของท่านเดินทางกันได้สะดวก แต่เนื่องจากแคว้นของท่านกว้างใหญ่ไพศาลเกินไป เอ็นหจิกจึงแต่งตั้งให้ punpun09 สหายคนสนิทของท่าน มาเป็นรัฐมนตรีการบิน โดย punpun09 คิดวิธีในการจัดระเบียบแต่ละสนามบินได้ดังนี้

- กำหนดให้สนามบินที่  $i$  มีตำแหน่งบนเส้นจำนวนเต็ม  $p_i$  และแทนด้วยรหัสตัวอักษรภาษาอังกฤษพิมพ์ใหญ่ 3 ตัว  $c_i$  โดยรับประกันว่าไม่มี 2 สนามบินใดๆ อยู่บนตำแหน่งเดียวกันหรือมีรหัสเหมือนกัน
- แต่ละสนามบินจะมี Gate ให้ทั้งหมด  $g_i$  Gate โดยเรียงลำดับเป็นเส้นตรงจาก Gate ที่ 0 ถึง  $g_i - 1$  โดยบาง Gate อาจมีเครื่องบินจอดอยู่ (ไม่ว่า)



punpun09 รัฐมนตรีการบิน  
อดีตกับตันมากฝีมือ

และได้ให้รหัสแทนแต่ละเที่ยวบิน ยกตัวอย่างเช่น  $CNX[0]$   $BKK[1]$  หมายถึงเครื่องบินออกจากสนามบิน CNX ที่ Gate หมายเลข 0 และอยากจะลงจอดที่สนามบิน BKK ที่ Gate หมายเลข 1 โดยมีรหัสการลงจอดและเงื่อนไขดังนี้:

1. **ไม่มีรหัส** ถ้าสามารถลงจอดที่ Gate ของสนามบินที่ต้องการได้
2. **ROUND** ถ้าสามารถลงจอดที่ Gate ของสนามบินที่ต้องการได้ และสนามบินต้นทางคล้ายปลายทาง
3. **SHIFT** หากไม่สามารถลงจอดที่ Gate ที่ต้องการได้ จะเลือก Gate ที่ว่างที่ใกล้ที่สุด หากมีหลาย Gate ใกล้เท่ากัน จะเลือกที่หมายเลขน้อยกว่า เช่น ถ้ามีอยู่ 5 Gate เริ่มจาก 0 ถึง 4 โดยที่ Gate 1 - 3 ไม่ว่าง แต่ Gate 0 กับ 4 ว่างทั้งคู่ และต้องการจอดที่ Gate 2 เราจะเลือก Gate 0 (0 กับ 4 ห่างจาก 2 เท่ากัน แต่  $0 < 4$ )
4. **ROUND\_SHIFT** เงื่อนไขเหมือน SHIFT แต่กรณีที่สนามบินต้นทางคล้ายปลายทาง
5. **CANCEL** หากทุก Gate ที่สนามบินนั้นเต็มหมด เที่ยวบินนี้จะ Cancel และกลับมาที่ต้นทาง

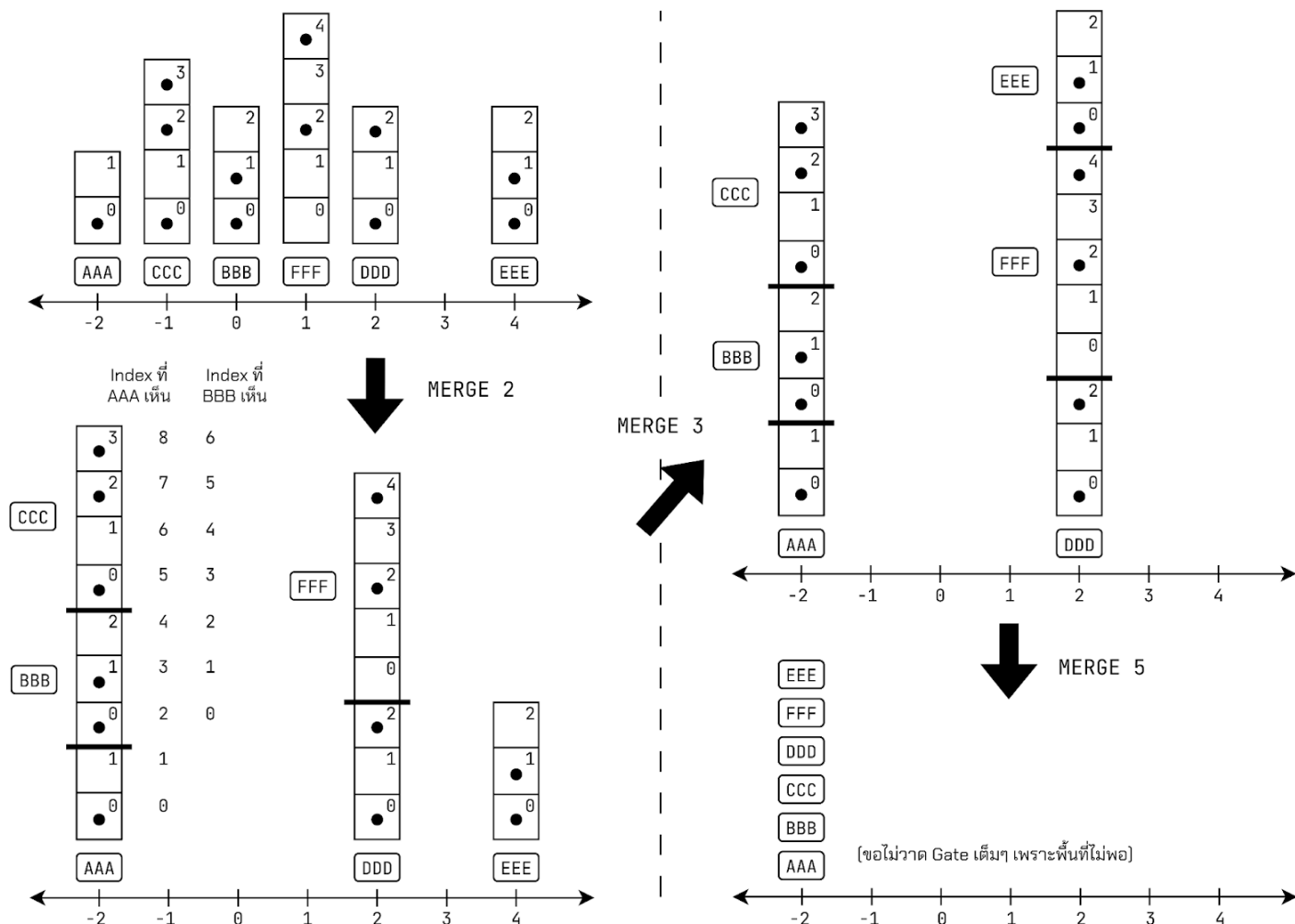
หมายความว่าหากเครื่องบินลงจอดได้สำเร็จ Gate ที่ต้นทางจะว่างลง โดยจะถือว่า Gate ต้นทางว่างก่อนที่เครื่องจะลงจอด

แต่ช่วงโคร้าย ดันเกิดอุบัติเหตุ โควิด-19 ทำให้ประชาชนต้องกักตัว เพื่อไม่ให้สนามบินที่อุตสาหะสร้างขึ้นมาต้องขาดดุล คุณ punpun09 จึงออกนโยบายให้มีการ “รวม” สนามบินที่ใกล้กัน ในระหว่างการบิน อาจมีคำสั่งให้รวมสนามบิน โดยมีนิยามดังนี้:

**MERGE K** หมายถึง ให้จับสนามบินเป็นกลุ่ม ซึ่งสำหรับทุกสนามบินในกลุ่ม **ระยะห่างจากสนามบินนั้นกับสนามบินที่อยู่ซ้ายสุดของกลุ่มมีค่าไม่เกิน K** เมื่อเริ่มจับกลุ่มจากทางซ้าย โดยแต่ละสนามบินสามารถอยู่ได้เพียง 1 กลุ่ม และเป็นไปได้ที่บางกลุ่มมีเพียง สนามบินเดียว การรวมสนามบินของแต่ละกลุ่ม จะนำ Gate ของแต่ละสนามบินในกลุ่มมาต่อกัน โดยลำดับที่นำมาต่อจะเรียงตามลำดับตามพจนานุกรมจากน้อยไปมากของสนามบิน

\*\* นิยามของ *ความคล้ายของสนามบิน* จะขอฟุดถึงในข้อสังเกตในตัวอย่างด้านล่าง

ตัวอย่างเช่น มีสนามบิน AAA, BBB, CCC ถึง FFF ซึ่งมีตำแหน่งที่ -2, -1, 0, 1, 2, 4 ตามลำดับ และมี Gate ดังรูป เมื่อทำการ MERGE 3 ครั้งโดยใช้ K เป็น 2, 3 และ 5 ตามลำดับ



จากภาพตัวอย่างให้สังเกตว่า

1. การจับกลุ่มจะไล่จากซ้ายสุดก่อนเสมอ สังเกตว่าที่จริง CCC, BBB และ FFF สามารถจับกลุ่มแล้วระยะไม่เกิน 2 (ค่า K ค่าแรก) จาก CCC (สนามบินซ้ายสุดของกลุ่มได้) แต่ไม่เลือก เพราะยึดจากทางซ้ายเป็นหลัก AAA จึงเป็นซ้ายสุดแทน
2. การเรียง Gate ของกลุ่ม CCC, AAA, BBB จะเรียงตามลำดับตัวอักษร โดยข้อมูลของแต่ละ Gate ยังคงอยู่
3. เมื่อต่อ Gate กันแล้ว การอ้างอิง Gate ใด Gate หนึ่งสามารถทำได้หลายวิธี โดยสนามบินที่อยู่ก่อนจะสามารถเห็น Gate ของสนามบินที่มาต่อทีหลังได้ เช่น หลังจาก MERGE 2 จะได้ว่า AAA[6], BBB[4], CCC[1] คือ Gate เดียวกัน จะนิยามคู่สนามบินว่า **คล้ายกัน** ก็ต่อเมื่อ สามารถเขียน Gate ของสนามบินหนึ่งในรูปของอีกสนามบินหนึ่งได้  
เช่น สามารถเขียน CCC[m] ในรูป BBB[m + 3] ; เมื่อ m เป็นจำนวนเต็ม 0 ถึง 3 แต่ไม่จำเป็นต้องทำในทางกลับกันได้ก็ได้  
จะได้ว่าสนามบิน CCC คล้ายกับ BBB และ BBB ก็คล้ายกับ CCC และถือแต่ละสนามบินจะคล้ายกับตัวเองด้วย
4. เมื่อจะรวมสนามบินที่เคยรวมมาแล้ว (เช่น หลังจาก MERGE 3) ลำดับการรวมจะขึ้นอยู่กับลำดับตามตัวอักษรของสนามบินหน้าสุดแต่ละกลุ่ม เช่น กลุ่มสนามบิน DDD, FFF จะรวมกับสนามบิน EEE โดยที่สนามบิน EEE จะต่อท้าย เพราะนำ DDD เทียบกับ EEE

ให้เขียนโปรแกรมเพื่อช่วย punpun09 คำนวณผลลัพธ์ของแต่ละเที่ยวบิน

## ข้อมูลนำเข้า

- บรรทัดแรก** รับจำนวนเต็ม  $A$  และ  $F$  แทนจำนวนสนามบิน และจำนวนเที่ยวบินตามลำดับ ( $1 \leq A \leq 10, 3 \leq F \leq 30$ )
- $A$  บรรทัดต่อมา** รับค่า  $c_i, p_i$  และ  $G_i$
- $c_i$  แทนรหัสแทนสนามบินที่  $i$  (" $AAA$ "  $\leq c_i \leq$  " $ZZZ$ ")
  - $p_i$  แทนรหัสแทนสนามบินที่  $i$  ( $-10 \leq p_i \leq 10$ )
  - $G_i$  แทนสายอักขระประกอบด้วยเพียง 1 หรือ 0 ที่แทน Gate ของสนามบินที่  $i$  โดย 1 ในตำแหน่งที่  $j$  ในสายอักขระ หมายความว่า Gate หมายเลข  $j$  ไม่ว่าง  $G_i$  มีความยาวไม่เกิน 20 ตัวอักษร สำหรับทุก  $i = 1, 2, \dots, A$
- $F + M$  บรรทัดต่อมา** เมื่อ  $M$  เป็นจำนวนคำสั่ง MERGE (ไม่รู้ค่า  $M$  ล่วงหน้า) โดยรับสายอักขระที่เป็นไปได้ 2 รูปแบบ
- $s_i[sg_i] d_i[dg_i]$  แทนเที่ยวบินจากสนามบิน  $s_i$  ที่ Gate  $sg_i$  ที่อยากไปยังสนามบิน  $d_i$  ที่ Gate  $dg_i$  โดยรับประกันว่าเป็นเที่ยวบินที่ถูกต้องเสมอ (หมายเลข Gate ไม่เกิน แต่เปลี่ยนในรูปสนามบินที่คล้ายกันได้ และ Gate ต้นทางจะมีเครื่องบินอยู่เสมอ)
  - MERGE  $K_i$  แทนคำสั่งในการ Merge ( $1 \leq K_i \leq 10$ )
- สำหรับทุก  $i = 1, 2, \dots, F + M$

## ข้อมูลส่งออก

มีได้มากที่สุด  $F$  บรรทัด โดยจะไม่แสดงข้อมูลเที่ยวบินที่ สามารถลงจอดที่ Gate ที่ต้องการได้ และสนามบินต้นทางไม่คล้ายกับปลายทาง นอกนั้นให้แสดงข้อมูลเที่ยวบินแต่ละบรรทัดในรูปแบบนี้

$i \quad code_{landing} \quad r_i[rg_i]$

- เมื่อ  $i$  เป็นหมายเลขเที่ยวบิน (เรียงจากลำดับ input)
- $code_{landing}$  เป็นรหัสการลงจอดของเที่ยวบินนั้น
- $r_i[rg_i]$  เป็น Gate ปลายทางที่เครื่องบินได้ลงจอด โดยจะแสดงแบบใช้ Gate ของสนามบินหน้าสุดเป็นหลัก ตัวอย่างเช่น (อิงจากภาพในหน้า 2 หลังจากการ MERGE 2) หาก Gate ที่ได้คือ CCC[2] จะต้องแสดง AAA[7] แทน

โดยหาก  $code_{landing}$  เป็น CANCEL จะไม่แสดง  $r_i[rg_i]$

และถ้าทุกเที่ยวบินลงจอดตามที่ต้องการได้หมด จะแสดงข้อความ NO CHANGE แทน

## ตัวอย่าง

```
2 5
RRE -1 0111
COL 3 1011
RRE[3] COL[3]
RRE[2] COL[2]
RRE[1] RRE[3]
COL[1] RRE[0]
RRE[0] RRE[2]
```

```
0 SHIFT COL[1]
1 CANCEL
2 ROUND RRE[3]
4 ROUND_SHIFT RRE[1]
```

6 6 AAA -2 10 CCC -1 1011 BBB 0 110 FFF 1 00101 DDD 2 101 EEE 4 110 AAA[0] CCC[1] EEE[0] FFF[2] MERGE 2 BBB[1] CCC[3] MERGE 3 DDD[2] EEE[0] BBB[6] DDD[2] FFF[5] AAA[5]	1 SHIFT FFF[1] 2 ROUND_SHIFT AAA[4] 3 ROUND DDD[8] 5 SHIFT AAA[3]
5 26 KNI 0 101101011 GHT -3 011011011101 CEM 2 11011101110 SVD 1 1011001110101 RRE -1 010110101 GHT[2] RRE[3] KNI[0] GHT[8] CEM[3] RRE[3] SVD[3] RRE[0] GHT[4] RRE[2] SVD[2] RRE[5] MERGE 3 KNI[3] RRE[3] CEM[1] SVD[1] KNI[2] CEM[12] GHT[5] CEM[9] RRE[6] SVD[1] KNI[13] SVD[3] GHT[1] CEM[9] SVD[0] CEM[5] RRE[2] SVD[2] KNI[7] CEM[16] MERGE 4 RRE[7] CEM[5] KNI[14] CEM[2] KNI[5] SVD[1] GHT[7] CEM[8] MERGE 6 GHT[7] CEM[8] CEM[12] GHT[14] SVD[8] RRE[3] KNI[2] KNI[1] KNI[1] CEM[37] KNI[8] GHT[22]	0 SHIFT RRE[2] 1 SHIFT GHT[6] 2 SHIFT RRE[5] 4 SHIFT RRE[7] 5 CANCEL 6 ROUND_SHIFT GHT[18] 7 ROUND CEM[12] 8 SHIFT CEM[10] 9 SHIFT CEM[6] 10 SHIFT CEM[14] 11 SHIFT CEM[15] 12 SHIFT CEM[3] 13 ROUND_SHIFT CEM[2] 14 SHIFT CEM[11] 16 SHIFT CEM[1] 17 SHIFT CEM[20] 18 SHIFT CEM[22] 19 CANCEL 20 ROUND_SHIFT CEM[24] 21 ROUND CEM[38] 22 ROUND_SHIFT CEM[47] 23 ROUND CEM[37] 24 ROUND CEM[37] 25 ROUND_SHIFT CEM[44]

## ข้อมูลชุดทดสอบ

มี 2 กรณีทดสอบที่ไม่มีการ MERGE และไม่มีการลงจอตแบบ SHIFT

มี 5 กรณีทดสอบที่ไม่มีการ MERGE และสามารถลงจอตได้ทุกรูปแบบ

มี 4 กรณีทดสอบที่มีการ MERGE เพียง 1 ครั้ง และสามารถลงจอตได้ทุกรูปแบบ

อีก 4 กรณีทดสอบไม่มีเงื่อนไขอื่นใด

Author: enderpalm

ข้อนี้เนื้อหาไม่เกิน Final แต่ยากกว่าข้อสอบจริง ทำไม่ได้ไม่เป็นไรครับ :)