

ส่งของในองค์กร (orgdelivery)

สำหรับคนที่ชอบซื้อสินค้าเป็นชีวิตจิตใจ การวางแผนว่าจะให้คนส่งสินค้ามาให้อย่างไรเป็นกิจกรรมแสนจะเพิลิตเพลิน องค์กรแห่งหนึ่ง มีแต่มนุษย์ชอบซื้อของ แถมมนุษย์ลูกน้องยังชอบส่งของให้เจ้านาย ช่างเป็นองค์กรสวรรค์สำหรับนักช้อปเสียนี้กระไร

องค์กรแห่งนี้มีพนักงาน N คน เรียกเป็นพนักงาน 0 ถึง $N - 1$ พนักงาน 0 เป็นหัวหน้าใหญ่สุด สำหรับพนักงาน i ใด ๆ จะมีหัวหน้าแผนกที่ขึ้นตรงด้วยคือพนักงาน $P[i]$ ส่วนพนักงาน 0 มี $P[0] = -1$ เราจะกล่าวว่าพนักงาน j เป็นหัวหน้าของพนักงาน i ถ้า j เป็นหัวหน้าที่ขึ้นตรงของ i (นั่นคือ $P[i] = j$) หรือ j เป็นหัวหน้าของ $P[i]$ ความสัมพันธ์หัวหน้าเป็นความสัมพันธ์แบบที่ไม่มีวงรอบ นั่นคือจะไม่มีใครเป็นหัวหน้าของตัวเอง

การส่งของในบริษัทนี้ จะมีการดำเนินการสองขั้นตอน พนักงาน a เมื่อต้องการส่งของให้พนักงาน b จะต้องดำเนินการดังนี้

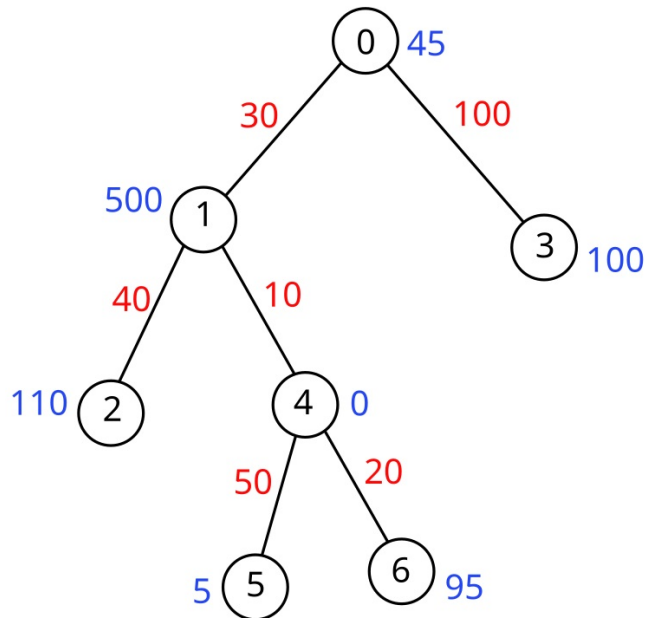
- **ขั้นแรก:** a สามารถเลือกพนักงาน c คนใดก็ได้ในบริษัท จากนั้น a จะส่งของให้ c (หมายเหตุ c อาจจะเป็น b เลยก็ได้)
- **ขั้นที่สอง:** ถ้า c ไม่ใช่เป้าหมาย c จะส่งของต่อไปยังหัวหน้าแผนกโดยตรง ($P[c]$) ซึ่งจะส่งต่อไปให้หัวหน้าของหัวหน้าไปเรื่อย ๆ จนกระทั่งถึง b (นั่นคือในการส่งจาก a ไปยัง b นั้น พนักงาน c ที่ a ส่งให้ในขั้นต้นแรก จะต้องมี b เป็นหัวหน้าด้วย)

วิธีการส่งของดังกล่าว มีค่าใช้จ่ายดังนี้

- **ขั้นแรก:** พนักงานทุกคนจะมีค่าระดับความเชี่ยวชาญการช้อป กล่าวคือ พนักงาน i จะมีระดับความเชี่ยวชาญ $V[i]$ ในการส่งขั้นแรกจาก a ไป c จะมีค่าใช้จ่าย $|V[a] - V[c]|$
- **ขั้นที่สอง:** เมื่อพนักงาน i ส่งของให้หัวหน้าโดยตรง $P[i]$ จะมีค่าใช้จ่าย $W[i]$

ค่าใช้จ่ายทั้งหมดจะเท่ากับค่าใช้จ่ายขั้นแรกรวมกับค่าใช้จ่ายขั้นที่สอง

พิจารณาตัวอย่างต่อไปนี้ ที่ $N = 7$



แผนผังแสดงความสัมพันธ์หัวหน้าลูกน้อง เส้นเชื่อมชั้นที่ติดกันคือเป็นหัวหน้า จำนวนเต็มสีน้ำเงินคือระดับความเชี่ยวชาญ $V[i]$ จำนวนเต็มสีแดงบนเส้นเชื่อมแทน $W[i]$ กล่าวคือค่าต่าง ๆ แสดงตามตารางด้านล่าง

i	0	1	2	3	4	5	6
$P[i]$	-1	0	1	0	1	4	4
$V[i]$	45	500	110	100	0	5	95
$W[i]$	0	30	40	100	10	50	20

สมมติว่าพนักงาน 3 ต้องการส่งของให้กับพนักงาน 1 ด้านล่างเป็นค่าใช้จ่ายถ้าเลือก c เป็นพนักงานคนต่าง ๆ สังเกตว่า $V[3] = 100$

c	0	1	2	3	4	5	6
ชั้นแรก	55	400	10	0	100	95	5
ชั้นสอง	-	0	40	-	10	60	30
รวม	-	400	50	-	110	155	35

ดังนั้นค่าใช้จ่ายต่ำที่สุดคือ 35

พิจารณาตัวอย่างที่สอง สมมติ 4 ต้องการส่งให้กับ 0 สังเกตว่าถ้าชั้นแรกส่งให้กับ 0 เลยจะเสียค่าใช้จ่าย $|0 - 45| = 45$ แต่จะไม่เสียชั้นที่สอง ใดๆก็ตามวิธีที่ดีที่สุด คือในชั้นแรก 4 ส่งให้กับตัวเอง เสียค่าใช้จ่าย $|45 - 45| = 0$ จากนั้นส่งต่อไปเป็นทอด ๆ ให้กับ 1 และ 0 มีค่าใช้จ่าย $10 + 30 = 40$ รวมเสียค่าใช้จ่ายสองชั้นเท่ากับ 40 ซึ่งน้อยที่สุดที่เป็นไปได้

พิจารณาอีกกรณีหนึ่ง สมมติ 1 ต้องการส่งให้กับ 5 สังเกตว่าชั้นแรก 1 จะส่งไปหา $c = 5$ ได้เพียงคนเดียวเท่านั้น เพราะว่า 5 ไม่ได้เป็นหัวหน้าใครเลย ดังนั้นค่าใช้จ่ายจะเท่ากับ $|V[1] - V[5]| = 495$

สำหรับข้อนี้คุณจะได้รับคำถามจำนวน Q คำถาม ที่แต่ละคำถามระบุว่าพนักงาน a ต้องการส่งของให้พนักงาน b สำหรับแต่ละคำถามให้คุณหาค่าใช้จ่ายน้อยที่สุดในการส่งของจาก a ไป b เป็นเท่าใด

รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันต่อไปนี้:

```
vector<long long> mincost(int N, vector<int> P,
                        vector<long long> V, vector<int> W,
                        int Q, vector<int> A, vector<int> B)
```

- N : จำนวนพนักงาน
- สำหรับ $0 \leq i < N$, พนักงาน i มี $P[i]$ เป็นหัวหน้าโดยตรง, $P[0] = -1$
- สำหรับ $0 \leq i < N$, $V[i]$: ระดับความเชี่ยวชาญของพนักงาน i
- สำหรับ $1 \leq i < N$, $W[i]$: ค่าใช้จ่ายถ้า i ส่งของให้ $P[i]$ ในชั้นที่สอง, $W[0] = 0$
- Q : จำนวนคำถาม
- สำหรับ $0 \leq j < Q$, คำถามที่ j ถามว่าค่าใช้จ่ายน้อยที่สุดที่พนักงาน $A[j]$ จะส่งของไปยังพนักงาน $B[j]$ เป็นเท่าใด
- ฟังก์ชันนี้จะต้องคืนค่าเป็นอาร์เรย์ขนาด Q โดยที่ข้อมูลตัวที่ j ในอาร์เรย์จะเป็นคำตอบของคำถามที่ j

เงื่อนไข

- $1 \leq N \leq 100\,000$
- $1 \leq Q \leq 100\,000$
- $0 \leq P[i] < N$, สำหรับ $i \neq 0$, $P[0] = -1$
- $0 \leq V[i] \leq 1\,000\,000\,000\,000$
- $0 \leq W[i] \leq 1\,000\,000$
- $0 \leq A[i] < N$
- $0 \leq B[i] < N$

ปัญหาย่อย

1. (5 points) $N \leq 1\,000$, $Q \leq 1\,000$
2. (5 points) $P[i] = 0$ สำหรับ $1 \leq i < N$
3. (12 points) ในชั้นที่สอง พนักงานคนใด ๆ สามารถส่งของให้ถึงพนักงาน 0 โดยส่งต่อไม่เกิน 10 ทอด
4. (12 points) $V[i] \in \{0, 100\,000\,000\,000\}$, $\sum W[i] \leq 90\,000\,000\,000$
5. (12 points) $P[i] = i - 1$ สำหรับ $1 \leq i < N$
6. (54 points) ไม่มีเงื่อนไขเพิ่มเติมอื่น ๆ

ตัวอย่าง

จากตัวอย่างข้างต้น เกรดเดอร์จะเรียกฟังก์ชัน mincost ดังนี้

```
mincost(7, [-1, 0, 1, 0, 1, 4, 4],  
         [45, 500, 110, 100, 0, 5, 95],  
         [0, 30, 40, 100, 10, 50, 20],  
         3, [3, 4, 1],[1, 0, 5])
```

ฟังก์ชันจะต้องคืนค่า [35, 40, 495]

เกรดเดอร์ตัวอย่าง

เกรดเดอร์ตัวอย่างจะอ่านข้อมูลนำเข้าดังนี้:

- บรรทัด 1: N Q
- สำหรับ $0 \leq i < N$, บรรทัด $2 + i$: $P[i]$ $V[i]$ $W[i]$
- สำหรับ $0 \leq j < Q$, บรรทัด $3 + N + j$: $A[j]$ $B[j]$

เกรดเดอร์พิมพ์ผลลัพธ์จากฟังก์ชัน mincost บรรทัดละจำนวน รวม Q บรรทัด

ข้อจำกัด

- Time limit: 1 seconds
- Memory limit: 512 MB