Network marketing

(1 sec, 512mb)

ส.อ. บวรทัต เด่นดำรงกุล หรือ “ TA Icy ” ที่เราทุกคนต่างรู้จัก นอกจากเค้าจะเป็น TA ที่มีความสามารถแล้ว หลายๆท่านยังรู้จักเขาในฐานะของ “ เส้นเลือดใหญ่ของวงการ Competitive Programming ประเทศไทย ” แม้ว่าจะเป็นโปรแกรมเมอร์ Tier S แต่แค่การเขียนโปรแกรมอย่างเดียวไม่สามารถทำให้เขามาได้ถึงจุดนี้ ส.อ. บวรทัต มีแหล่งรายได้หลักนั่นก็คือ ธุรกิจเครือข่าย ที่สร้างรายได้ให้กับเค้าอย่างมหาศาล

เมื่อเวลาผ่านไปนานเข้า ส.อ. บวรทัต ได้รู้ตัวว่าธุรกิจของเขานั้นมันได้เติบโตจนเกินควบคุมไปแล้ว ดังนั้นในฐานะโปรแกรมเมอร์อัจฉริยะ เขาจึงได้ออกแบบโปรแกรมเพื่อที่จะช่วยในการควบคุมธุรกิจของเขาได้ง่ายมากยิ่งขึ้น และในฐานะ “ เส้นเลือดใหญ่ของวงการ Competitive Programming ประเทศไทย ” เขาจึงได้ใช้อำนาจของเขาในการบังคับคุณให้เขียนโปรแกรม ดังกล่าวให้เค้า

แนวคิดหลักภายใต้ธุรกิจเครือข่ายก็คือ หากลูกข่าย pi (สมาชิกคนใดคนหนึ่งในเครือข่าย) สามารถหาลูกค้า ci ของตัวเองได้ pi จะมีสถานะเป็น parent ของ ci และ ci จะตกเป็น child ของ pi และเป็น ลูกข่ายคนใหม่ในเครือข่ายของ ส.อ. บวรทัต ทันที

หากก่อนหน้านี้ pi เป็น child ของสมาชิกคนใดๆและสมาชิกคนนั้นมี parent ของตัวเองทั้ง สมาชิกคนนั้น และ parent ของ สมาชิกคนนั้น รวมถึง parent ของ parent ของ สมาชิกคนนั้น เป็นลำดับไปเรื่อยๆจนถึง parent ที่ไม่มีสถานะเป็น child ทุกคนในขั้นบันใดนี้จะถือว่าได้ ci เป็น child ของตัวเองเพิ่มทั้งหมด

ตัวอย่างเช่น ส.อ. บวรทัต เป็น parent ของ TA spade และ TA spade เป็น parent ของ TA best หากวันหนึ่ง TA best สามารถชวน TA Tack มาเป็นลูกค้าได้ทั้ง ส.อ. บวรทัต, TA spade และ TA best จะถือว่าได้ TA Tack เป็น child ของตนเอง หลังจากนั้นหาก TA Tack สามารถหาลูกค้า ของตนเองได้ทุกชื่อที่กล่าวมาข้างต้นก็จะได้รับ child เพิ่ม

อย่างไรก็ดี หาก TA spade สามารถชวน TA worralop มาเข้าร่วมเครือข่ายได้จะมีเพียง ส.อ. บวรทัต และ TA spade ที่ได้ child เพิ่มเพราะมีเพียงผู้ที่หาลูกค้าได้และ parent ของเขาเท่านั้นที่จะได้รับ child เพิ่ม

เนื่องจากในธุรกิจนี้ไม่ได้มีเครือข่ายของ ส.อ. บวรทัต เพียงแค่เครือข่ายเดียวดังนั้นมีความเป็นไปได้ที่ลูกค้าคนใดๆก็ **อาจมีเครือข่ายของตนเองอยู่แล้ว** ก็เป็นไปได้ ตัวอย่างเช่น หาก TA DungeonOwner มี child อยู่แล้ว 3 คนหลังจากนั้น ส.อ. บวรทัต สามารถชักชวน TA DungeonOwner มาเป็น child ของเค้าได้ ในกรณีนี้ ส.อ. บวรทัต จะได้ child เพิ่ม 4 คน (TA DungeonOwner และ child ของ TA DungeonOwner ทั้งสามคน)

งานที่ต้องทำ

จงเขียน template <typename T> class network_marketing ที่มี T เป็น datatype ของ รหัสแทนตัวลูกข่ายแต่ละคน โดยจะมีทำฟังก์ชั่นสำคัญสองอย่างได้แก่

- size_t count_children(T x) ฟังก์ชั่นนี้จะทำหน้าที่ในการคืนจำนวน child ของลูกข่ายที่รหัสแทนที่เป็น x
- void setparent(T child,T parent) ฟังก์ชั่นนี้จะทำหน้าที่ในการ set ให้ลูกข่ายที่มีรหัสแทนที่ด้วย child มี parent เป็นลูกข่ายที่มีรหัสแทนที่ด้วย parent โดยรับประกันว่าจะไม่มีการเรียกฟังก์ชั่นนี้ด้วยค่า child ที่ซ้ำกันอย่างแน่นอน

ข้อบังคับ

- โจทย์ข้อนี้จะมีไฟล์ตั้งต้นมาให้ ซึ่งประกอบด้วยไฟล์ network_marketing.h และ main.cpp อยู่ ให้นิสิตเขียน code เพิ่มเติมลงในไฟล์ network_marketing.h เท่านั้น และการส่งไฟล์เข้าสู่ระบบ grader ให้ส่งเฉพาะไฟล์ network_marketing.h เท่านั้น
    - ไฟล์ network_marketing.h จะต้องไม่ทำการอ่านเขียนข้อมูลใด ๆ ไปยังหน้าจอหรือคีย์บอร์ด หรือไฟล์ใด ๆ
- หากใช้ VS Code ให้ทำการ compile ที่ไฟล์ main.cpp

**** main ที่ใช้จริงใน grader นั้นจะแตกต่างจาก main ที่ได้รับในไฟล์โปรเจ็กต์เริ่มต้น แต่จะทำการทดสอบในลักษณะเดียวกัน ****

คำอธิบายฟังก์ชัน main

main() จะสร้าง CP::network_marketing<int> tree ขึ้นมา แล้วรับคำสั่งดังนี้

- s ตามด้วย integer c และ integer p เพื่อทำการเรียก tree.setparent(c,p)
- c ตามด้วย integer x เพื่อทำการเรียก tree.count_children(x) และพิมพ์ค่าออกทางหน้าจอ
- q เพื่อจบการทำงาน

ชุดข้อมูลทดสอบ

- 5% n = 3 10%
- n,m <= 10 และรับประกันว่า ค่า b ใน set-parent ไม่ซ้ำกันเลย
- n,m <= 10 และรับประกันว่า ค่า b ใน set-parent ซ้ำกันไม่เกิน 1 ครั้ง
- 15% n,m <= 1000 และรับประกันว่า ค่า b ใน set-parent ซ้ำกันไม่เกิน 1 ครั้ง
- 15% n,m <= 1000

- 45% ไม่มีข้อจำกัดอื่นใด

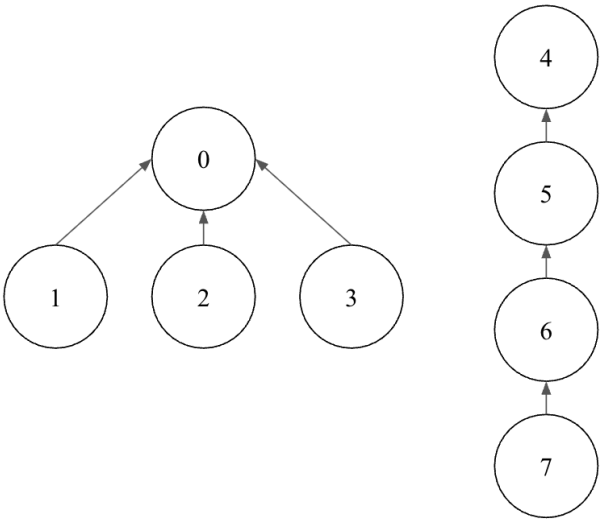
คำแนะนำ (สำคัญ)

ในการจะได้คะแนนเต็มนั้น void setparent(T child,T parent) ต้องทำงานเสร็จใน O(N) และ size_t count_children(T x) ต้องทำงานเสร็จใน O(1) เท่านั้น

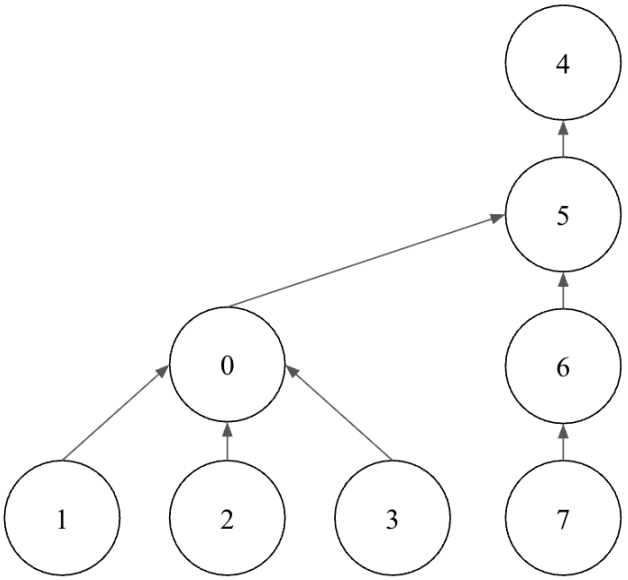
ตัวอย่างการทำงานของ main

ข้อมูลนำเข้า	ข้อมูลส่งออก
s 1 0 s 2 0 s 3 0 c 0 c 3 c 4 q	3 // 0 มี child 3 คนได้แก่ 1, 2 และ 3 0 // 3 ไม่เคยโดนเรียก setparent โดนที่ตัวเองเป็น parent จึงมีจำนวน child ค่าเป็น 0 0 // 4 ไม่เคยอยู่ในเครือข่ายนี้จึงมีจำนวน child ค่าเป็น 0
s 1 0 s 2 0 s 3 0 s 5 4 s 6 5 s 7 6 c 0 c 4 c 5 s 0 5 c 4 c 5 c 6 q	3 // 0 มี child 3 คนได้แก่ 1, 2 และ 3 3 // 4 มี child 3 คนได้แก่ 5, 6 และ 7 2 // 5 มี child 2 คนได้แก่ 6 และ 7 7 // หลังคำสั่ง s 0 5 ลูกข่าย 4 มี child 7 คนได้แก่ 5, 6, 7, 0, 1, 2 และ 3 6 // หลังคำสั่ง s 0 5 ลูกข่าย 5 มี child 7 คนได้แก่ 6, 7, 0, 1, 2 และ 3 1 // 6 มีเพียง 7 เป็น child

คำอธิบายเพิ่มเติมตัวอย่างที่สอง



หลังจากการทำงานใน 6 บรรทัดแรกเราจะสามารถแสดงเครือขายของเราได้ดังนี้ตัวเลขใน node คือรหัสแทนที่ของลูกข่ายแต่ละคนและลูกษรจะชี้จาก child ไปที่ parent หลังจากมีคำสั่ง s 0 5 กราฟของเราจะมีหน้าตาดังนี้



0, 1, 2 และ 3 จะกลายเป็น child ของ 5 และ 4 ทำให้ผลการเรียก child หลังคำสั่ง s 0 5 เป็นดังที่ปรากฏในตัวอย่าง