Whops!-logger

Tjenesten Whops! lar brukere overføre penger mellom hverandre gjennom en app. De prøver i så stor grad som mulig å unngå å lagre informasjon om brukerene sine sentralt, og lar heller informasjonen lagres på brukeren sin egen mobiltelefon.

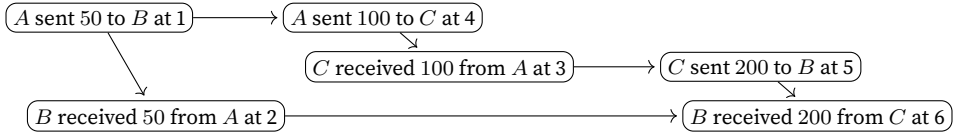
En gang i blant har de behov for å anskaffe en komplett logg av overføringer for en periode for å kunne feilsøke systemene sine. Da henter de inn *lokale* logger fra noen utvalgte mobiltelefoner i systemet. En lokal logg er en liste av hendelser, der hver hendelse beskriver enten at et beløp er sent eller mottatt. I tillegg inneholder hver hendelse et tidsstempel som forteller når beløpet ble sendt eller motatt. For enkelhets skyld er tidsstemplene representert som positive heltall. Her er et lite eksempel på tre logger fra mobiltelefoner A, B og C.

A	B	C
A sent 50 to B at 1	B received 50 from A at 2	C received 100 from A at 3
A sent 100 to C at 4	B received 200 from C at 6	C sent 200 to B at 5

Dessverre er tidsstemplene fra mobiltelefonene upålitelige. De observerer ofte at et mottak av et beløp forekommer *før* beløpet ble sendt, i følge tidsstemplene. I eksempelet ovenfor mottar C et beløp på 100 *før* A har sendt beløpet til C.

For å approksimere en riktig logg lar de hver hendelse fra de lokale loggene være noder i en rettet graf $G = (V, E)$. Dersom to hendelser u og v kommer direkte etter hverandre i en lokal logg, så finnes det en rettet kant (u, v) i grafen. For hver hendelse u som repre-

senterer at et beløp er sendt, så finnes det en rettet kant (u, v) der v er hendelsen for det korresponderende mottaket av det beløpet. Her er grafen for eksempelet ovenfor:



Den ønskede komplette loggen for dette eksempelet skal se slik ut:

A sent 50 to B at 1
B received 50 from A at 2
A sent 100 to C at 4
C received 100 from A at 3
C sent 200 to B at 5
B received 200 from C at 6

Hver node v kan aksessere tidsstempelet fra loggen ved $v.\texttt{ts}$. Skriv en algoritme som tar rettet graf $G = (V, E)$ som input, og skriver ut alle noder i V i en rekkefølge slik at:

- Dersom det finnes en kant fra u til v så skal u skrives ut før v.
- u skal skrives ut før v hvis $u.\texttt{ts} < v.\texttt{ts}$, så lenge det ikke bryter med det første kravet.