Rundtur med Den Norske Turistforeningen (DNT)

Fagutvalget ved institutt for informatikk har bestemt seg for å arrangere en rundtur i et vakkert norsk fjellandskap for alle IFI-studenter. Dette sammenfaller med at Den Norske Turistforeningen har lansert en ny app, sammen med offentlige datasett, slik at det er gode muligheter for å gjøre et godt informert rutevalg. Siden de fleste studenter på IFI har en forkjærlighet for grafer, er det ikke til å unngå at de vil bruke datasettet til å bygge en graf som fanger nok informasjon til å kunne algoritmisk bestemme den beste rundturen.

Grafen som bygges er en rettet graf $G = (V, E)$, der hver node $u \in V$ representerer en lokasjon, og en kant $(u, v) \in E$ representerer en turvei fra u til v. Ingen noder har en kant til seg selv.

(a) Det viser seg at grafen har en ganske spesiell egenskap: *Hver node inngår i makismalt én sykel*. Hver sykel representerer en *rundtur*.

Gi en algoritme som finner alle rundturene i grafen, og oppgi kjøretidskompleksiteten. Du kan kalle på algoritmer kjent fra pensum.

(b) En av rundturene fra (a) blir valgt som reisemål, hvor t er en node i rundturen. Turen starter ved et *utfartssted*, som er enkelt å komme seg til med kollektivtransport. Et utfartssted er identifisert ved noder som har inngrad 0. Det siste som gjenstår i planleggingen er å velge et egnet utfartssted turen skal starte ved for å nå t.

Datasettet inneholder heldigvis også informasjon om hvor lang tid en gjennomsnittlig turgåer bruker langs en turvei. Dette blir gitt som en vektfunksjon w, slik at for hver kant $(u, v) \in E$, så angir $w(u, v)$ tidsbruk som et positivt heltall. Vi vil bruke vektene til å finne ut hvilket utfartssted som er best egnet for å komme seg til reisemålet.

Input er en rettet og vektet graf $G = (V, E)$ med vektfunksjon w, en mengde med utfartssteder U og en node t i reisemålet. Output er noden $s \in U$ slik at avstanden mellom s og t er så lav som mulig.

Gi en algoritme som effektivt finner utfartsstedet som har lavest avstand til t i reisemålet, og oppgi kjøretidskompleksiteten. Du kan kalle på algoritmer kjent fra pensum.