Blindern-problemet 10 poeng

Blindern-problemet er en utfordring som er kjent for mange studenter ved Universitet i Oslo, som oppstår når man må gå fra en forelesning til en annen på motsatt side av campus på et knapt kvarter. Du er blitt bedt om å konsultere i UiO sitt nye prosjekt hvor det skal utvikles et tunnelsystem som forbinder alle bygningene som hører til campus.

Det er allerede kartlagt hvor mye det vil koste å grave tunnel mellom hvert par av bygninger (og anta at ingen av de kartlagte tunnelene kolliderer). Din oppgave er å finne den mest kostnadseffektive måten å grave et tunnelsystem på, ut ifra de kartlagte tunnelene, slik at man kan gå mellom alle bygningene kun ved å bruke tunnelsystemet.

(a) Forklar kort hvordan dette problemet kan uttrykkes som et grafproblem. Svaret ditt bør inkludere:

- Hva slags graf (rettet/urettet og vektet/uvektet) egner seg her?
- Hva representerer nodene?
- Hva representerer kantene?

> Problemet bør tenkes på som en komplett, vektet og urettet graf, der hver node i representerer en bygning og hver kant representerer en tunnel som kan bygninges ut. Kantene er vektet etter hvor mye det koster å grave tunnelen.
>
> Opptil 2 poeng for en god forklaring av hva slags graf oppgaven handler om.

(b) Oppgi en egnet algoritme, kjent fra pensum, som kan brukes til å finne den mest kostnadseffektive måten å grave ut et tunnelsystem på, slik at man kan komme seg mellom alle bygningene kun ved å bruke tunnelsystemet. For algoritmen må du oppgi:

- de mest sentrale datastrukturene den bruker,
- en *kort* forklaring på hvordan algoritmen fungerer, og
- kjøretidskompleksiteten på algoritmen.

> Vi kan løse problemet ved å finne et minimalt spenntre over G. Algoritmen vi kjenner best fra kurset er Prim sin algoritme for minimale spenntrær.
>
> Den fungerer ved å bygge et tre fra en vilkårlig valgt startnode. Naboene av startnoden plasseres på en prioritetskø, der vekten på kanten utgjør prioriteten. Grafen traverseres videre ved å plukke noder fra prioritetskøen, og legge dem til i treet dersom noden ikke allerede er i treet.
>
> Når alle noder er i treet vil vi ha bygninget et minimalt spenntre for G.
>
> Algoritmen er i $\mathcal{O}((|V| + |E|) \cdot \log(|V|))$, som kan forenkles til $\mathcal{O}(|E| \cdot \log(|V|))$ fordi grafen er sammenhengende. Siden grafen er komplett kan kjøretidskompleksiteten også uttrykkes som $\mathcal{O}(|V|^2 \cdot \log(|V|))$. Alle disse gir like god uttelling.
>
> - Opptil 5 poeng på deloppgaven for en god forklaring av en effektiv forklaring av en algoritme for minimale spenntrær.
> - Opptil 2 poeng for datastrukturene.
> - Opptil 2 poeng for forklaringen.
> - 1 poeng for riktig kjøretidskompleksitet.

(c) Prosjektet blir ferdig i rekordfart og er klar til å brukes. Beskriv, med naturlig språk, en effektiv algoritme for å finne korteste sti fra en bygning til en annet kun ved å bruke tunnelsystemet. Du kan bruke algoritmer kjent fra pensum *uten* å gjøre rede for dem. Oppgi kjøretidskompleksiteten på algoritmen.

Det resulterende tunnelsystemet er et *tre*, som betyr at det kun finnes én sti mellom hvert par av noder. Derfor kan du finne stien i $\mathcal{O}(|V|+|E|)$, med en hvilken som helst traversering, for eksempel BFS eller DFS.

Å bruke Dijkjstra her er ikke helt veldefinert, fordi vektene på kantene som er beskrevet i oppgaven har med kostnaden av å bygninge tunnelene, ikke hvor lang tid det tar å gå i dem. Det vil riktignok gi riktig svar, fordi vektene ikke spiller noen rolle når det kun finnes én sti mellom hvert par av noder.

- Opptil 3 poeng for en løsning med BFS eller DFS.
- Opptil 1 poeng for en løsning med Dijkstra.
- 1 poeng for riktig kjøretidskompleksitet (for en løsning med Dijkstra må man ha riktig kjøretidskompleksitet for å få uttelling for deloppgaven).