

Algoritmer og Datastrukturer - opgave i grafer/aktivitetsplanlægning

I denne opgave skal der arbejdes med filer, klasser, objekter og vektorer/ArrayLists.

Lav en tekstfil ved navn `data.txt` med følgende indhold:

```
1;A;3
1;B;6
1;C;4
2;D;5
3;E;4
3;F;1
4;G;2
4;H;7
4;I;1
5;J;4
```

Den type af filer kaldes *kommaseparerede* filer og bruges ofte til logning af måledata og andre strukturerede data.

Her er der tale om data til en aktivitetsplan for et udviklingsprojekt med tre informationer pr aktivitet.

Den første information kalder vi *event*, og den angiver rækkefølgen af aktiviteterne; den anden kalder vi *task*, og den angiver navnet på aktiviteten; den sidste hedder *duration* og indeholder aktivitetens varighed i dage.

Opgaven går ud på følgende:

- Lav en klasse ved navn `Aktivitet`, som indeholder attributterne `event (int)`, `task (string)` og `duration (int)` med de sædvanlige metoder (constructors, get-metoder).
- Indlæs aktiviteterne fra filen og isoler attributterne.
- Hver aktivitet transformeres til et `Aktivitet`-objekt, og objekterne placeres i en `vector//ArrayList` ved navn `tabel`.
- Herefter udskrives antallet af aktiviteter (10) og den gennemsnitlige varighed som decimaltal (3,7 dage).
- Udførelsen af de enkelte aktiviteter er afhængig af attributten `event`. Fx kan task D (`event 2`) ikke udføres, før A, B og C (`event 1`) alle er udført, og task J (`event 5`) kan ikke udføres før G, H og I (`event 4`) er udført. Aktiviteter med samme `event`-nummer kan altså udføres parallelt/samtidigt.

- Der er således en række aktiviteter, der kan kaldes *den kritiske vej*, hvorom det gælder, at ingen af de indgående aktiviteter kan blive forsinkede, uden at det forsinker hele projektet. Fx kan task A godt blive forsinket, fordi task D alligevel ikke kan starte, før task B er færdig. Men en forsinkelse af B, forsinker hele projektet.
- Du skal udskrive varigheden af den kritiske vej, dvs. den tid det minimum tager at udføre projektet (26 dage) samt rækkefølgen af tasks (B, D, E, H, J). Opgaven skal løses ved at gennemløbe `vector/ArrayList` tabel, og ikke ved at læse inputfilen igen.
- Precondition: du kan antage, at events er i stigende orden startende med 1 og uden 'huller'. Tasks, derimod, kommer ikke altid nødvendigvis i stigende alfabetisk orden som i dette tilfælde.
- Endelig skal du lave en helt ny tekstfil (eller flere) med andre data, som du kan teste dit program yderligere med.

I forhold til hvordan programmeringsopgaver løses i industrien, er det af afgørende betydning, at du følger ovenstående specifikation til punkt og prikke. Fx ikke noget med at bruge andre navne på attributterne end de angivne eller at bruge et `array` i stedet for en `vector/ArrayList`.

Ole Dolriis, november 2024