

1. What is the time complexity of your solution? You must explain the stated complexity. If you have a different complexity for each problem, you need to discuss them separately.

- TopologicSort() har $O(|V| + |E|)$ kompleksitet. Dette fordi den utfører en topologisk sortering med en form for BFS i en stack med mellomlagring av noder før de sorteres.

Kompleksiteten har den tidskompleksiteten ettersom den itererer gjennom kantene og nodene i grafen kun en gang. Deretter legges de i stacken.

- Cyclicpath() har $O(|V| + |E|)$ kompleksitet utfører en DFS søk for å finne ut og kalkulere om grafen har en sykel med samme tidskompleksitet $O(|V| + |E|)$, ettersom den itererer gjennom nodene i grafen og tilhørende kanter en gang, helt fram til søket har gått gjennom totalt alle noder og finner en sykel.

- Print() har $O(n^2)$ kompleksitet. Den inneholder en for-løkke som itererer gjennom shortestAmountOfTime og er avhengig av dennes verdi, der n er shortestAmountOfTime med en nøstet løkke som itererer gjennom alle antall tasks i HashMappet MapOfTasks. shortestAmountOfTime er taskens første mulige tid addert med taskens tid.

- Hele løsningen vil uansett kjøre i $O(N^2)$ kompleksitet siden man tar utgangspunkt i den verst tenkelige kjøretiden, selv om søkingen etter sykel har en kjøretid på $O(|V| + |E|)$.

2. What requirements are there for the input graph in order for the project to be successfully planned? Please put the discussion in the light of graf theoretical properties.

- Grafen som leses inn må være en rettet graf eller en såkalt DAF graf. Grafen må ha ikke-negative kanter som forteller avhengigheten mellom nodene. Nodenes avhengighet benyttes i topologisk sortering. Cyclicpath() sjekker om grafen er syklisk.

3. What kind of the graph algorithms did you use in your implementation?

- Benyttet topologisk sortering og DFS algoritme. TopologicSort() legger taskene i prioritetskø i linjær rekkefølge. DFS benyttes til å sjekke om grafen har sykliske pather.