

1.

Kjøretid for `realizable()`: Her har vi brukt en algoritme for topologisk sortering, en form for BFS gjennomgått i forelesning, med kjøretid  $O(|V|+|E|)$ . Grafens noder og kanter besøkes en gang før de legges inn i stacken, derav kjøretiden nevnt over.

Kjøretid for `earlyStart()/lateStart()`: Både `earlyStart()` og `lateStart()` inneholder nøstede løkker, for å gå gjennom både listen med noder, og den enkelte nodes lister med avhengigheter. Derfor blir kjøretiden  $O(n^2)$ .

Kjøretid for programmet totalt: Selv om kjøretiden for topologisk sortering er  $O(|V|+|E|)$ , vil kjøretiden for hele programmet bli  $O(n^2)$ . Dette er fordi vi regner ut ifra et worst-case scenario, og  $O(n^2)$  er den tregeeste kjøretiden.

2.

For at programmet skal kunne kjøres, kreves det at grafen som sendes inn er en rettet, asyklisk graf. Dette vil si at grafens noder har avhengigheter knyttet til hverandre. Programmet krever dette, ettersom den topologiske sorteringen bruker avhengighetene til å plassere nodene i forhold til hverandre.

Vi kan også si at vi krever at grafen som sendes inn ikke har en sykel, men programmet håndterer dette i tilfelle det sendes inn en syklisk graf også.

3.

Vi bruker algoritmen for topologisk sortering, en form for BFS, for å sortere tasks etter avhengighet til hverandre. Denne lar oss fortsette med en liste som er sortert i rekkefølgen oppgavene skal gjennomføres.