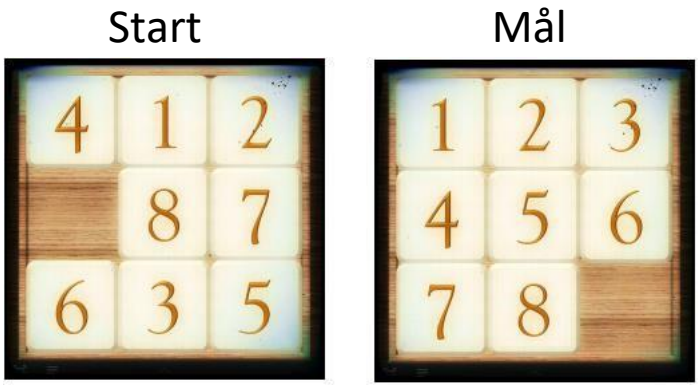
INFO 283 – Grunnleggende algoritmer i kunstig intelligens

Lab-oppgave 1 – Datoer: 9. og 12. september 2019

I denne oppgaven skal de få prøve å programmere litt i en enkel kunstig intelligens-applikasjon.

Programmet dere skal jobbe med er et program som finner løsninger for 8-spillet. Et lite brett med fire brikker som blir flyttet rundt. Figuren under viser spillet, hva som kan være en startposisjon og hva målet er.



På GitHub finner dere koden dere trenger: https://github.com/SonjerBolan/INFO283_V2019

Dere kan enten laste ned filene og importere de i et prosjekt eller dere kan klone repositoriet.

For kloning trenger dere en IDE som støtter Git. Seminarleder bruker PyCharm og vil dermed anbefal dere og gjøre det samme. (Går helt fint å bruke andre IDE'er også)

PyCharm: https://www.jetbrains.com/pycharm/download/#section=windows
Velg community edition!

Spør seminarleder om dere trenger hjelp!

I main-metoden i filen **__init__.py** er det lagt inn kode som kan teste 3 søkestrategier på ulike startposisjoner for 8-spillet. For å velge søkestrategi og startposisjon kan du ta vekk og legge til kommentar-tegn («#»). Programmet skriv ut en løsning på problemet, en kostnad på løsningen og hvor mange noder i problemgrafen som har blitt gjennomgått.

Programmeringsoppgave:

Det som mangler i programmet er implementering av metodene/funksjonene **move_left**, **move_rigth**, **move_up** og **move_down** i klassen **EightGameNode**. Disse skal ta hånd om flytting av den tomme ruten, som i programmet er indikert med verdien 0. For at programmet skal virke skikkelig må du programmere disse. Programmer disse og prøv ut om metodene/funksjonene dine

virker ved å kjøre MoveTest-klassen. Se at den tomme ruten flytter seg i henhold til hvilke funksjoner som blir kalt.

Analyse-oppgave:

Når flytte-programmet ditt virker kan du kjøre det på de 4 ulike start-bretta som fins i Main-klassen og på de ulike søkestrategiene. Hvor mange flytt i løsningene og hvor mange noder har blitt undersøkt i søket skrives ut. Det kan ta litt tid å få kjørt programmet, så vær tålmodig.

Best-først-strategien er ikke interessant i forhold til Bredde-først om den ikke har ulike kostnader på trekkene. Standardkost per trekk er nå 1. Implementer en versjon der du setter kostnaden til 10 dersom en flytter den blanke til ruten der 6-eren skal stå (board[2][0]). Prøv best-først med denne versjonen. Har det konsekvenser for noen av utgangsstillingene. Prøv og med andre kostnads-modeller (f. eks. det koster ekstra mye å flytte den tomme mot høyre).