Oblig 3 (med både del 1 og del 2)

Leveringsfristen er 17. november (og leveringsfristen for konkurransen vil være noen få dager senere).

Les først:

På del 1 under kan man jobbe to og to sammen. Begge skal i så fall levere løsningen, men skal oppgi hvem man har jobbet sammen med. De som er med i konkurransen og lager et program som går ut over det som er krevet i del 1, kan levere konkurranseprogrammet som svar på denne oppgaven. Del 2 av denne obligen skal hver student løse alene.

Del 1: Rett fram utgave av konkurranse-oppgaven

(Dette er mye den samme teksten som til konkurransen, men en del er fjernet)

Oppgaven er å skrive et program for løsing av "15-spillet", som også kommer som 8-spillet og generelt (NxN-1)-spillet (på et NxN-brett). Dette er diskutert i læreboka som "8-puzzle game" på side 717. Programmet skal lages generelt for NxN-brett, og data inn til programmet skal være først tallet N (på egen linje), og så N linjer med N tall i hver, som angir startposisjonen (der 0 angir det tomme feltet). Eksempel:

```
3
1 2 3
0 4 5
7 8 6
```

Programmet skal da finne frem til en måte å flytte brikkene slik at man kommer til målsituasjonen:

```
1 2 3
4 5 6
7 8 0
```

(og tilsvarende for NxN-brett). Løsningen man finner skal ha færrest mulig trekk. Et lovlig trekk er altså å la den tomme posisjonen (0-en) ”bytte” posisjon med en av nabobrikkene. Et slikt trekk angis med hvordan *den tomme posisjonen* flytter seg, med V, H, O, N (for Venstre, Høyre, Opp og Ned). En løsning på problemet over er derved: HHN, og denne skal skrives ut på skjermen om programmet finner en løsning. Programmet skal bruke A*-søking, og kan passelig bruke en rett fram Manhattan-heuristikk (se oppgave 23.7).

Denne metoden bør kunne løse alle 8-spill-problemer, og noe mer vil ikke bli forlangt. Det er jo imidlertid morsomt å se om det også kan løse enklere 15-spill-problemer (slike som kan løses i få trekk), og det er morsomt om dere legger ved i leveringa en kommentar om hva programmet klarer, og på hvilken tid.

Programmet skal få angitt et filnavn ved oppstart, og på den kan det ligge flere oppgaver etter hverandre, med en blank linje mellom. Fila avsluttes ved at N angis til null.

Del 2: Uavgjørbarehet og NP-kompletthet

Oppgave 1

Avgjør om påstandene under er riktige eller gale. Gi en kort forklaring.

a. Formelle språk (slik det er definert i dette kurset) er en formalisering av vanlige språk som for eks. Spansk og Engelsk.

b. Desisjonsproblemer er like komplekse som tilsvarende optimaliseringsproblemer.

c. Problemer som ikke kan løses i polynomisk tid kalles ”NP-komplette problemer”.

Oppgave 2

Hvilke av problemene under er uavgjørbare? Gi en fyllestgjørende forklaring. (Bruk gjerne anledningen til å lage en enkel Turingmaskinkonstruksjon og å angi en detaljert reduksjon fra Stoppeproblemet.)

a. $L_1 = \{0\}$

b. $L_2 = \{M \mid M \text{ gjenkjenner } L_1\}$

c. $L_3 = \{M \mid M \text{ gjenkjenner } L_2\}$

Oppgave 3

Bevis at 2-HAMILTONBARHET (definert under) er NP-komplett.

Definisjon av 2-HAMILTONBARHET: Gitt en sammenhengende graf G som input. Bestem om det finnes to enkle løkker i grafen som ikke har noen felles node, og som til sammen inkluderer alle nodene.