INF 4130 Oppgavesett 5, 04/10-2011 m/løsningsforslag

Oppgave 1

Hvordan ville du tilordnet «lovende-het» til noder i et tilstandsroms-tre i et best-først-søk (når de legges inn i prioritetskøen), slik at nodene tas ut igjen i samme rekkefølge som i et dybde-først-søk hhv. et bredde-først-søk.

For dybde-først søk, bruker vi en global teller *C* initiert til en meget høy verdi (MAXINT) og som minkes hver gang vi prosesserer en node.

For bredde-først holder det vel med en global teller *C´* initiert til 0, og som økes hver gang vi prosesserer en node.

Oppgave 2

Løs oppgave 23.6 fra læreboka (B&P).

Merk at det her er viktig at vi ikke teller med det tomme feltet når vi regner hva som er på gal plass. Om vi gjør det kan vi ikke vise det vi skal. Se for eksempel på 2x2-brettet:

3	1
	2

Her ser vi at vi kunne få alt på plass ved tre trekk, mens om vi regnet med det tomme feltet så ville vi få $h(v)$=4. Det essensielle er dermed at når vi gjør et trekk, så er det bare forflytningen av en ekte brikke (=«tile») som teller inn på heuristikk-verdien. Dermed er $h(v)$ ihvertfall ikke større enn antall nødvendige trekk fra *v*, og det er håp om monotonitet.

Vi ser først at *h*(sluttsituasjonen) er null, så den forutstningen er OK.

Vi ser så på en vilkårlig situasjon u, og et trekk fra denne til tilstedene v. Vi må vise at $h(u) + c(u,v) >= h(v)$, der $c(u,v)$=1, siden vi regner kosten i antall trekk. Vi må dermed vise at $h(v)-h(u) <= 1$, altså, at *h*-verdien ikke øker mer enn 1 når vi gjør et trekk. Siden det bare er én brikke som flyttes i ett trekk, blir dette helt selvfølgelig, siden det maksimalt er denne som kan flytte seg ut av riktig posisjon i dette trekket (og dermed øke *h*-verdien med 1).

Oppgave 3

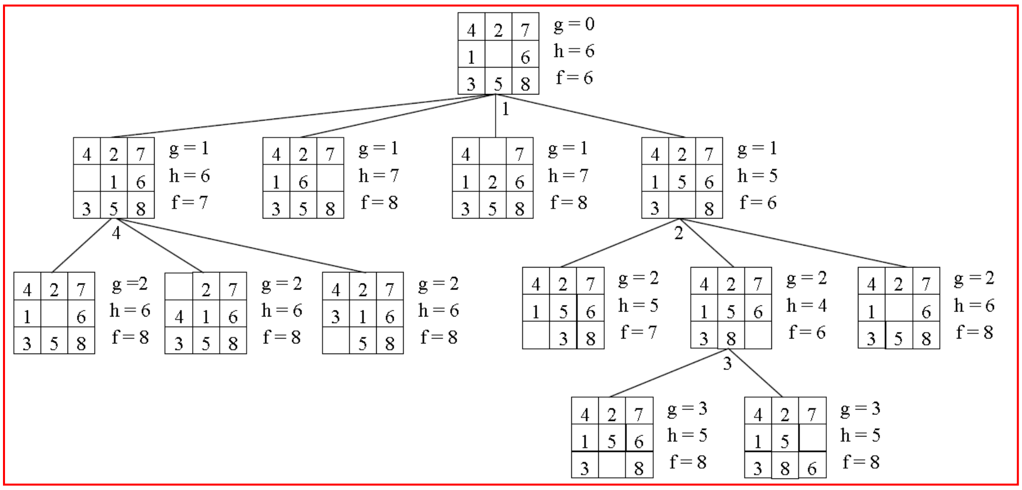
Løs oppgave 23.7 fra læreboka.

Når vi bare merker oss poenget fra forrige oppgave er også denne grei. Ved siden av at *h*(sluttsituasjonen) må være null (opplagt), må vi også her vise at *h*-verdien ikke øker med mer enn 1 i ett trekk. Siden bare én brikke flyttes, blir forandringen i dens Manhattan-avstand også forandringen i *h*-verdien. Men vi ser lett at om en brikke flyttes ett hakk horisontalt eller ett hakk vertikalt, blir forandringen i Manhattan-avstanden ikke mer enn 1, og dermed er vi i mål.

Oppgave 4

Løs oppgave 23.8 fra læreboka.

(Her med alle forbehold om trykkfeil.) I det fjerde trekket har vi to noder med f = 7, men vi forutsetter her en FIFO-disiplin for de med lik prioritet. I neste omgang blir det den andre med f = 7 som blir ekspandert videre.



Oppgave 5

Vil svaret ditt angående monotonisitet fra 23.7 også holde om vi tillater å flytte hullet diagonalt?

(Svarformuleringen antar at 23.7 allerede er løst). Summen av Manhattan-distansene vil nå ikke lenger være en monoton heuristikk, rett og slett fordi den kan bli større enn antall trekk som er nødvendig. Se på:

1	2	3
4		6
7	8	5

Her kan man løse oppgaven i ett trekk (bytte 5 og den tomme), mens Manhattan distansen vil gi h=2. Det er imidlertid lett å se at om en brikke er i (x1, y1) og dens riktige posisjon er (x2, y2), så blir færrest antall flytt til (x2,y2) (om den er alene på brettet) lik: max(|x2-x1|, |y2-y1|). Vi vil derfor forsøke med summen av denne tatt over alle brikkene, som heuristikk, og den er hvertfall slik at h(sluttsituasjonen) er null.

For å vise monotonitet må vi igjen vise at h-verdien ikke forandrer seg mer enn 1 når vi gjør et trekk. Siden det bare er én brikke som flyttes, koker spørsmålet ned til om max(a,b) kan forandre seg mer enn 1, når a kan gå én opp eller ned, og b tilsvarende (samtidig!). Litt tellig på fingrene sier en raskt at maksimal forandring er 1, og dermed er vi i mål.

Oppgave 6

Hva med å bruke den faktiske kostnaden som heuristikk? Vil den faktiske kostnaden alltid være monoton? Vil vi ekspandere et mindre tre? Hva er eventuelt problemet?

Altså, h(s) er her kosten av de gjenstående trekk langs en optimal vei fra tilstand s for å komme til en løsning. Det er selvfølgelig litt rart å tenke seg at man har dette som heuristikk, siden man da jo allerede har løst problemet. Men dette kan si noe om hvordan algoritmen vil virke om man har en veldig god heuristikk. Om h er slik som angitt over ser vi at verdien f(s) = g(s) + h(s), når tilstanden s taes ut av prioritetskøen og over i treet, er lik den korteste veien fra starttilstanden til en løsning, som går gjennom tilstanden s. Det er da ikke vanskelig å se at A*-algoritmen vil gå rett på den best mulige løsning og ikke "tulle" med andre løsninger. Dersom det er mange tilstander som har den samme f-verdi kan algoritmen imidlertid forfølge mange like lange veier i parallell, avhengig av i hvilken rekkefølge man velger like verdier fra prioriteskøen.

Oppgave 7 (om det er tid igjen)

Vis at lengden av den rette linje (eller egentlig storsirkel, men det behøver vi ikke tenke på) fram til målet er en monoton heuristikk for å finne korteste vei slik man gjør i kap 23.3.3 (side 728).

Anta at målet for reisen er *w,* da ser vi hvertfall at *h(w)=0*. Anta så at vi har to steder *u* og *v* som det går vei mellom, av lengde *c(u,v),* og som har direkte luftavstand *a(u,v).* Vi må vise *h(v)-h(u) <= c(u,v)*. Vi vet at *h(v)-h(u) <= a(u,v)* ut fra trekantulikheten, og det er også opplagt at *a(u,v) <= c(u,v)* (den rette linje er den korteste), og dermed er saken i orden.

Oppgave 8 (om det er tid igjen)

Sett på g-, h- og f-verdier på tilstandene i figur 23.7 (side727) og sjekk at vi faktisk får så stor avskjæring i forhold til fullt bredde-først-søk som ville gitt figur 23.3 (side 719).

Overlates til den enkelte.

[slutt]