

Løsningsforslag for utvalgte oppgaver fra kapittel 7

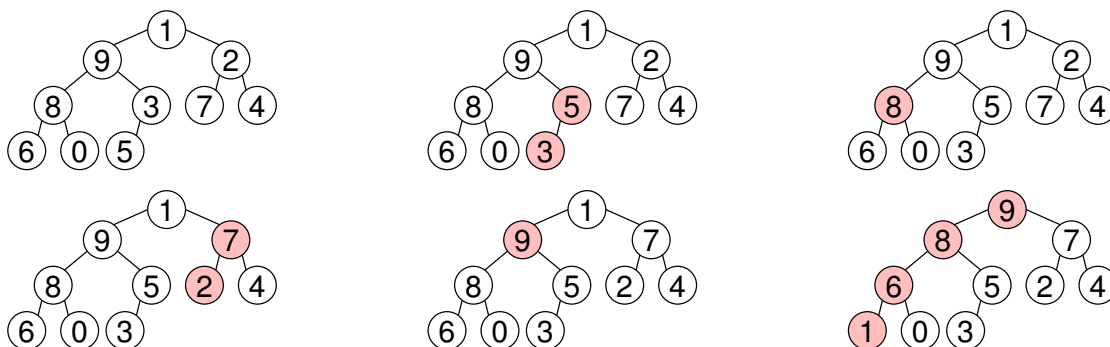
Innhold

7-1	Lage max-heap og min-heap	1
7-2	Vis at en sortert tabell er en heap	2
7-4	Eksempel på <i>hent_maks</i>	2
7-8	Er heapsort stabil?	2

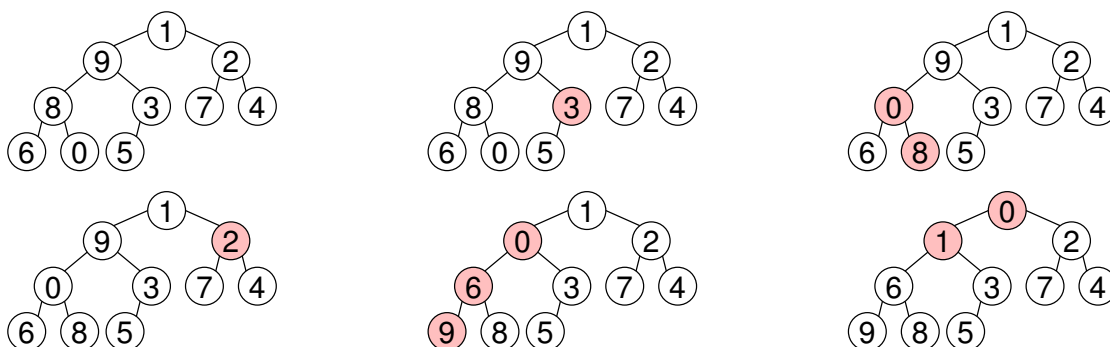
7-1 Lage max-heap og min-heap

Farge viser hvilke noder som flytter seg i hver omgang.

En max-heap blir til:



En min-heap blir til:



7-2 Vis at en sortert tabell er en heap

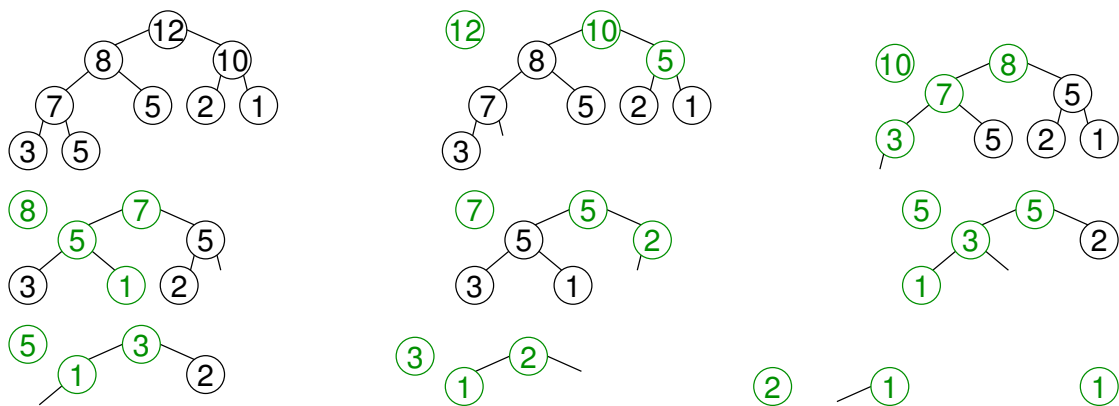
I en sortert tabell forekommer tallene i stigende orden. Jo høyere tabellindeks, jo høyere er tallet som ligger der.

Heap-egenskapen for en min-heap krever at barnenodene har høyere prioritet enn foreldrenoden. I følge formel 7.2 og formel 7.3 på side 141 er indeksen for en barnenode alltid høyere enn indeksen for noden selv, den er nemlig alltid en eller to mer enn dobbelt så høyt. Dermed ser vi at barnenoder alltid ligger på en høyere indeksposisjon i tabellen, og siden tabellen er sortert finner vi alltid en høyere verdi der. Så kravet til heapegenskapen er oppfylt!

7-4 Eksempel på hent_maks

Farge viser hvilke noder som flytter seg i hver omgang. Vær oppmerksom på at det er flere riktige løsninger.

Her fjerner vi resten av nodene fra max-heapen på side 146 i læreboka:



7-8 Er heapsort stabil?

Heapsortering er ikke en stabil sorteringsalgoritme. Hvis vi sorterer en heap med bare like tall, vil ikke *lag_heap()* bytte om noen av tallene. Men den etterfølgende løkka i heapsort vil sette det første tallet inn på den siste tabellposisjonen. Dermed ble ikke rekkefølgen bevart.