Innsetting i heap 10 poeng

(a) Du er gitt et tomt array på størrelse 7.

0	1	2	3	4	5	6

Sett inn de følgende tallene med min-heap innsetting:

$$3, 5, 10, 8, 4, 2, 7$$

Fyll ut tabellen slik den ser ut etter alle tallene er satt inn. Skriv svaret som en kommaseparert liste.

Tallene settes inn slik:

3						
3	5					
3	5	10				
3	5	10	8			
3	5	10	8	4		
3	4	10	8	5		
3	4	10	8	5	2	
3	4	2	8	5	10	
2	4	3	8	5	10	7
0	1	2	3	4	5	6

Det er kun den siste linjen `2, 4, 3, 8, 5, 10, 7` som trenger inngå i svaret.

2 poeng for riktig svar. 1 poeng for *nesten* riktige svar, som vil si dersom

- svaret er en min-heap, men ikke den samme heapen som den gitte rekkefølgen vil gi opphav til, eller
- svaret er ett bytte unna løsningensforslaget.

(b) Forklar kort hvordan algoritmen for innsetting i en min-heap fungerer, og skisser algoritmen med pseudokode. Du kan anta at arrayet i input ikke er fullt.

Input: Et array `A` som representerer en heap med n elementer, og et element x
Output: Et array som representerer en heap, som inneholder x

```
1 Procedure HeapInsert(A, x)
  | // ...
```

Hovedidéen er å alltid legge til på neste ledige plass, det vil si indeks n. Det er der neste node *må* være for at treet fortsatt skal være komplett. Hvis noden på den nye plassen er mindre enn foreldrenoden så må de bytte plass. Dette gjøres iterativt (eller rekursivt).

I arrayrepresentasjonen av en heap ligger alltid roten på indeks 0. Foreldrenoden til en gitt indeks i ligger på `ParentOf(i)` $= \lfloor \frac{i-1}{2} \rfloor$ for en gitt indeks i.

Her er algoritmen presentert med pseudokode:

```
Input: Et array A som representerer en heap med n elementer, og et element x
Output: Et array som representerer en heap, som inneholder x
1 Procedure HeapInsert(A, x)
2     A[n] ← x
3     i ← n
4     while 0 < i and A[i] < A[ParentOf(i)] do
5         A[i], A[ParentOf(i)] ← A[ParentOf(i)], A[i]
6         i ← ParentOf(i)
7     return A
```

Det gis opptil 6 poeng for et godt svar på oppgaven.

- Opptil 1 poeng for å gi hvor elementet skal plasseres.
- Opptil 2 poeng for å få frem at noden skal bytte plass med foreldernoden dersom den er mindre.
- Opptil 3 poeng for en godt skissert algoritme.

(c) Redgjør for kjøretidskompleksiteten til algoritmen.

Algoritmen for innsetting i heap er $\mathcal{O}(\log(n))$. Det er fordi n halveres i hver iterasjon.

Det gis 1 poeng for riktig kjøretidskompleksitet og 1 poeng for redegjørelsen.