Litt om prioritetskøer og binære heaps

Med *heap* mener vi arrayer som representerer binære min-heaps.

(a) En array med n elementer kan gjøres om til en heap i $\mathcal{O}(n)$.

(b) En min-heap blir en max-heap ved å reversere arrayet.

(c) Alle elementer på dybde d i en heap er mindre enn alle elementer på dybde $d+1$.

(d) Innsetting i en heap med n elementer er $\mathcal{O}(\log(n))$ i verste tilfelle.

(e) Et AVL-tre kan brukes som en prioritetskø med samme kjøretidskompleksitet som en heap.

(f) Nodene langs en sti fra rotnoden til en løvnode i en heap er ordnet fra minst til størst.

(g) Dersom vi bytter plass på to søskennoder i en heap, er det fremdeles en heap.

(h) En prioritetskø med konstant tid på alle operasjoner ville gjort at Dijkstra hadde kjøretidskompleksitet $\mathcal{O}(|V|+|E|)$.

(i) Man kan finne det *største* elementet i en *min*-heap i $\mathcal{O}(\log(n))$.

(j) I en heap ligger *over halvparten* av elementene på de *to dypeste* nivåene.