

## Oblig 8: Heap-ordnede binære trær

Oppgave D:

### 1. Når treet er komplett

- Metoden `lagHeapOrdning()` kalles rekursivt på venstre og høyre subtre, som betyr at hvert nivå av treet behandles.
- Etter å ha behandlet begge subtrærne, kalles `reparer()`, som i verste fall kan gå fra roten og helt ned til bladene, noe som gir en kjøretid på  $O(\log n)$ .
- Hver node blir besøkt én gang og kan forårsake en  $O(\log n)$  reparasjon, som gir en total kjøretid på  $O(n \log n)$

### 2. Når hver node (bortsett fra én) har nøyaktig ett subtre

- I dette tilfellet er treet nesten en lineær kjede, hvor hver node kun har ett barn.
- `lagHeapOrdning()` kalles rekursivt nedover hele treet, noe som gir en rekursjonsdybde på  $O(n^2)$
- `reparer()` vil i verste fall også gå gjennom hele den lineære strukturen, som gir en total kjøretid på  $O(n^2)$