

Svarark — Post 6 (facit)

Øvelse 1 — Find element

Hvor mange personer skulle I sammenligne med for at finde den rigtige?

Svar: varierer — afhænger af hvor i træet personen befinder sig. Typisk få sammenligninger.

Hvad ville worst case være hvis I var 1.000 personer i træet?

Svar: ca. 10 sammenligninger — du halverer antallet af kandidater ved hvert trin ($\log_2 1000 \approx 10$).

Hvad er kompleksiteten? **$O(\log n)$**

Sammenlign med Post 2 (find specifikt element) — hvad er forskellen?

I Post 2 (ArrayList) skulle du spørge alle fra starten — $O(n)$. I træet halverer du antallet af kandidater ved hvert trin — $O(\log n)$. Med 1.000 elementer er forskellen 1.000 sammenligninger vs. ca. 10.

Øvelse 2 — Slet element

Hvad skete der med personerne under den slettede node?

Det afhænger af om noden har børn. Hvis ingen børn: bare fjern. Hvis ét barn: det overtager pladsen. Hvis to børn: find den nærmeste efterfølger (mindste i højre undertræ) og sæt den på pladsen.

Hvad er kompleksiteten for at finde og slette et element i træet? **$O(\log n)$**

Øvelse 3 — Indsæt element

Hvor mange sammenligninger skulle I lave for at finde den rigtige plads?

Svar: varierer — én sammenligning pr. niveau i træet.

Hvad er kompleksiteten? **$O(\log n)$**

Ville det gøre en forskel hvis alle i træet havde fødselsdag i januar? Hvad ville det gøre ved træets form — og ved kompleksiteten?

Træet ville degenerere til en lang kæde mod venstre — i princippet en LinkedList. Kompleksiteten ville falde til $O(n)$ fordi du aldrig halverer antallet af kandidater. Det er præcis problemet balancerede træer (AVL, Red-Black) løser.