IN2010, høsten 2023

«Obligatorisk» oppgave 4

Hashing og kattiser

Oppgave 1: Hashing

a) Ligger vedlagt. Kjøres med Kommando:

```
$ python HashingMenKindaIkke.py < mengd/inputs/input_100000 |
 cmp - mengd/outputs/output_100000
```

Dette er i bunn og grunn en direkte implementasjon av Linear Probing, rasket fra notatene til Lars. Jeg valgte å gå for LP fordi det skal være noe raskere enn SC, og den kjørte jevnt over bedre for meg. En ting jeg stusset på og som jeg slet lenge med å fikse helt til jeg fikk et tips av en kompis var at telleren ikke oppførte seg i det hele tatt. Tror det har noe med rehashing å gjøre, for trikset var å bare øke N (størrelsen på det tomme arrayet) til resultatene gav mening, som for meg ikke gir mening.

b) Funn:

Til min overraskelse opplever jeg at hashing med LP faktisk går *litt* raskere enn BST med 100k inputs og en fornuftig valgt (vilt gjettet) N. Søking i AVL utgår selvfølgelig fullstendig. Lavere inputs er det ikke merkbar forskjell. Ordbøker funker!

Oppgave 2: Kattunge

Kjøres via

```
$ python KattisFastIEtTre.py < katt/inputs/input_5 | cmp - ka
tt/outputs/output_5
```

Ganske rett frem. Lager et «tre»…altså, det er jo allerede laget for oss, vi leser det inn og lagrer det i en ordbok med lister hvor nøkkelen er foreldrenoden og innholdsverdien er en mengde med dets barn. Funksjonen `finnvei()` starter først med å finne hvilken foreldrenode som er over kattenoden, og hopper til denne foreldrenoden. Den oppdaterer så kattenoden og gjentar prosessen helt til den sitter på rotnoden. Alle stegene blir lagt til i en liste som vi printer ut på et fint format sånn at den kan sammenlignes med forventet ut.