Algoritmer og Datastrukturer #5

16.09.19 – Hashtabeller

* Formål
  + Rask gjenfinning av lagrede verdier (nøkler)
  + Rask innsetting av nye verdier (nøkler)
  + Selv om nøkkelen ikke kan brukes som tabellindeks
* Virkemåte
  + Vanlig tabell bruker nøkkelen som indeks: tabell[nøkkel]
  + Noen nøkler er uegnet som indeks:
    - tabell[«tekst»]
    - tabell[0.93]
  + Hashtabeller bruker en *hashfunksjon*.
    - Funksjonen tar nøkkelen som parameter, og produserer en tabellindeks
    - tabell[hashfunksjon(nøkkel)] 🡨 Beregner indeksen
* Anvendelser
  + Finne igjen elementer i Θ (1) tid, selv om nøkkelen *ikke* er egnet som tabellindeks.
    - Sortert tabell med binærsøk blir *O* (log *n*)
  + Indeks i databaser, gir raskere oppslag
  + Oppslag i symboltabeller i kompilatorer
  + Stavekontroll
  + Finne frem i filsystemet i et OS
  + Java: alle klasser har en hashCode(), en hashfunksjon som beregnes fra objektets innhold. Brukes av HashMap
* Gode hashfunksjoner
  + Må alltid gi en verdi mellom 0 og tabellstørrelsen
  + Går *raskt* å beregne
  + Gir god *spredning*
  + Gir få *kollisjoner*. (Ulike nøkler *hasher* til samme posisjon)
* Åpen adressering
  + Når elementer kolliderer, legger vi de ekstra elementene andre steder i tabellen. (Enkleste er «neste ledige plass»)
  + Tabellen kan ikke overfylles
  + Det lønner seg å ikke fylle hele tabellen, men ha ~25% overhead.