

Ugeseddel: Hashing

Eva Rotenberg*

19. april 2022

Om denne uge

Materialer *Introduction to Algorithms* (CLRS): kap. 11 på nær 11.5.

Opgaver

1 Håndkøring og egenskaber

- 1.1 Indsæt følgende nøglesekvens $K = 7, 18, 2, 3, 14, 25, 1, 11, 12, 1332$ i et array af størrelse 11 vha. hægtet hashing med hashfunktionen $h(k) = k \bmod 11$.
- 1.2 Indsæt følgende nøglesekvens $K = 2, 32, 43, 16, 77, 51, 1, 17, 42, 111$ i et array af størrelse 17 vha. lineær probering med hashfunktionen $h(k) = k \bmod 17$.
- 1.3 Slet 111 og 51 fra arrayet produceret i forrige delopgave.
- 1.4 Antag at vi laver sletning i lineær probering *uden* at genindsætte elementer i klynge til højre for det slettede element. Giv en kortest mulig sekvens af ordbogoperationer der viser at dette ikke virker korrekt.
- 1.5 Lad K være en sekvens af nøgler gemt i et array A vha. hægtet hashing. Givet A , kan man effektivt finde det maksimale element i K ?

2 Divisorer i divisionmetoden

Overvej følgende hashfunktion $h(k) = k \bmod 10$ og kig på nøglesekvensen $K = 0, 5, 20, 40, 65, 15, 90, 95, 80, 55$.

- 2.1 Hvad er problemet med valget af hashfunktion i forhold til K ?
- 2.2 Diskuter hvorfor man benytter primtal i divisionsmetoden.

3 Doven sletning i lineær probering

Overvej følgende “dovne” strategi til sletning i lineær probering. Når et element slettes på position p , markerer vi blot at indholdet på position p er slettet.

- 3.1 Beskriv, hvordan SEARCH og INSERT skal modificeres med denne strategi.
- 3.2 Diskuter fordele og ulemper i forhold til “ivrig” sletning.

*baseret på materiale af Bille&Gørtz

4 Spilserverstatistik

Til dit nye, meget succesfulde onlinespil vil du gerne holde styr på, om den megen aktivitet kommer fra en lille gruppe spillere, der ofte spiller spillet, eller mange forskellige spillere, der kun spiller spillet sjældent. Hver spiller har et unikt ID og du kan i din spilserver tilgå sekvensen af spiller-ID'er fra alle spil.

4.1 Giv en algoritme, der tæller antallet af unikke spillere på spilserveren.

4.2 Giv en algoritme, der finder den spiller, der har spillet flest spil.

5 Bitvektorer

Almindelige computere betegnes ofte som w -bit computere, hvor hver lagerplads består af w bits (fx $w = 64$). Man kalder en lagerplads også et *ord* af w bits. Det betyder, at registre og lagerceller kan rumme w bits, og at de fleste datatyper (for f.eks. heltal, flydende tal, og referencer,) er repræsenteret som w -bit ord. De fleste gængse programmeringssprog understøtter operationer på bitniveau på ord af w bits, herunder forskydninger (*shift*) og bitvise logiske operationer. En *bitvektor* er et array af bits (0er og 1er). Antag, at vi har en w -bit computer og løs følgende opgaver.

5.1 Vis hvordan man kompakt kan repræsentere en bitvektor B af længde w , således at man i $O(1)$ tid kan tilgå eller ændre værdien af den i 'te bit i B .

5.2 Vis hvordan man kompakt kan repræsentere en bitvektor B af længde n (hvor n er meget større end w), således at man i $O(1)$ tid kan tilgå eller ændre værdien af den i 'te bit i B .

5.3 Vis hvordan man kan bruge en bitvektor til kompakt at repræsentere en dynamisk mængde uden satellitdata ved hjælp af direkte adressering.

6 * Sortering af små tal

Lad $A[0..n-1]$ være et array af heltal fra $\{0, \dots, n-1\}$. Giv en algoritme til at sortere A i $O(n)$ tid. *Hint*: start med at sætte tallene ind i et array med hængt hashing og identitetsfunktionen som hashfunktion.

7 ** Uinitialiserede arrays

Vi vil gerne implementere et kæmpe array A så vi effektivt kan hente og ændre værdien på et element. Til at starte med kan elementerne indeholde "skrald" og pga. af størrelsen vil vi ikke bruge tid på at initialisere alle elementerne. Giv en løsning, der bruger lineær plads i størrelsen af arrayet, kan hente eller ændre værdien af en indgang i $O(1)$ tid og kun skal bruge $O(1)$ tid på initialisering. *Hint*: vedligehold en stak af størrelse svarende til antallet af elementer der ikke er skrald i A . Vedligehold pointere til og fra A til hurtigt at afgøre om et element er skrald eller data.

8 Algorithm Awards

Hjælp Josefine med at tælle stemmer til den årlige algoritmiske prisuddeling. (Se CodeJudge.)