Ugeseddel: Hashing

Philip Bille

Om denne uge

Litteratur *Introduction to Algorithms*, Cormen, Rivest, Leisersons og Stein (CLRS): kap. 11 pånær 11.5.

Opgaver

1 Håndkøring og egenskaber

1.1 [o] Indsæt følgende nøglesekvens $K = 7, 18, 2, 3, 14, 25, 1, 11, 12, 1332$ i tabel af størrelse 11 vha. af hægtet hashing med hashfunktionen $h(k) = k \mod 11$.

1.2 [o] Indsæt følgende nøglesekvens $K = 2, 32, 43, 16, 77, 51, 1, 17, 42, 111$ i tabel af størrelse 17 vha. linæer probering med hashfunktionen $h(k) = k \mod 17$.

1.3 Slet 111 og 51 fra tabellen produceret i opgave 1.2.

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 ordbogoperationers, der viser at dette ikke virker korrekt.

1.5 Lad K være en sekvens af nøgler gemt i en tabel A vha. hægtet hashing. Givet A kan man effektivt finde det maksimale element i K?

2 Divisorer i divisionmetoden Overvej følgende hashfunktion $h(k) = k \mod 10$ 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 Forklar hvorfor vi 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 Forklar hvordan SEARCH og INSERT skal modificeres med denne strategi.

3.2 Forklar fordele og ulemper i forhold til "ivrig" sletning.

4 Bitvektorer En *bitvektor* er en tabel af bits (0'er og 1'er).

4.1 Vis hvordan man kompakt kan repræsentere en bitvektor B af længde n således man i $O(1)$ tid kan hente eller ændre værdien af den i'te bit i B. *Hint:* tænk i bitwise operationer som shift, and og or.

4.2 Vis hvordan man kan bruge en bitvektor til kompakt at repræsentere en dynamisk mængde uden satellit data vha. af direkte addressering.

5 Spilserverstatistik Til dit nye meget succesfulde online spil 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 ID'ere fra alle spil.

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

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

6 [$*$] **Sortering af små universer** Lad $A[0..n-1]$ være en tabel af heltal fra $\{0,\ldots,n-1\}$. Giv en algoritme til at sortere A i $O(n)$ tid. *Hint:* start med at sætte tallene ind i en hægtet hashtabel med identitetsfunktionen som hashfunktion.

7 [$**$] **Uinitialiserede tabeller** Vi vil gerne implementere en *kæmpe* tabel A så vi effektivt kan hente og ændre værdien på en indgang. Til at starte med kan indgangene indeholde "skrald" og pga. af størrelsen vil vi ikke bruge tid på initialisere alle indgangene. Giv en løsning, der bruger lineær plads i størrelsen af tabellen, 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 pegere til og fra A til hurtigt at afgøre om et element er skrald eller data.