

SI-möte #12, Programkonstruktion och datastrukturer

Elias Castegren

elca7381@student.uu.se

22 februari 2011

Begrepp

- i) Vad är en *hashtabell*? Vad gör en hashfunktion? Vad är hashfunktionens definitions- och värdemängd?
- ii) Vad menas med *uniform hashing*? Vad innebär *perfekt hashing*? Vilka villkor måste vara uppfyllda för att perfekt hashing ska vara möjligt?
- iii) Hur definieras en hashtabells *loadfaktor*? Vad innebär *rehashing*?
- iv) Vad innebär *chaining* när man pratar om hashtabeller?
- v) Vad är *öppen adressering (open addressing)*? Hur fungerar *linjär* och *kvadratisk probing*? Vad innebär *clustering*? Hur fungerar *dubbel hashing*?
- vi) Hur många element kan man lagra i en hashtabell med m platser om man använder chaining respektive open addressing? Vilka värden kan loadfaktorn anta med de olika teknikerna?

Övningar

1. Antag att du har en hashtabell med 10 celler som använder chaining. Tabellens hashfunktion är

$$h(k) = k \bmod m$$

där m är antalet celler i hashtabellen. Sätt in följande element:

113, 19, 155, 16, 129, 43, 99, 26, 125, 73

Ta sedan bort elementen

19, 26, 129, 73

Vad händer om man försöker leta upp elementet 33? När vet man att man kan sluta leta?

2.

Sätt in och ta bort samma element som i uppgift 1, fast i två hashtabeller som använder öppen adressering med linjär respektive kvadratisk probing (utan rehashing) istället för chaining. Använd samma hashfunktion som ovan. För den kvadratiske probingen kan du använda probfunktionen $f(i) = i^2$. Räkna det totala antalet prober som behövs när man sätter in och tar bort elementen från de olika tabellerna.

3.

Sätt in elementen 112, 63, 48 i en hashtabell med samma hashfunktion som ovan och $m = 11$ celler. Använd kvadratisk probing med probfunktionen $f(i) = i + i^2$. Försök hitta elementet 90. När kan man sluta leta? Sätt sedan in 91 och 90 i hashtabellen. Ta bort 91 och leta efter 90 igen. Vad är skillnaden från förra gången?

4.

Vad är den maximala listlängden för en hashtabell med chaining med m celler i bästa och värsta fall om loadfaktorn är

i) 0,8 ii) 1,2 iii) 5,8?

5.

Sätt in elementen

16, 21, 26, 31, 36, 41, 46

i två hashtabeller med längd $m = 5$ och samma hashfunktion som ovan. Använd linjär och kvadratisk probing (med probfunktionen $f(i) = i^2$). Rehasha vid behov genom att skapa en ny hashtabell med längd $2m$ och föra över de gamla elementen. Gör sedan om samma sak men rehasha till $2m - 1$. Jämför hur många prober som behövs vid rehashing.

Lycka till!