

Øving 5 - Algoritmer og Datastrukturer

Henrik Halvorsen Kvamme

6. oktober 2023

1 Introduksjon

Denne øvingen har 2 deloppgaver som må gjøres. Den første ber deg lese fra en fil med navn til alle som tar aldat faget for å legge inn navnene som nøkler til et hashmap med lenka lister.

Deloppgave 2 ber deg lage to implementasjoner av hashmap med unike tall som nøkler. Implementasjonene bruker to former for adressering: lineær probering og dobbel hashing. For et den skal godkjennes må m være minst 10 000 000. Man må også teste ulike fyllingsgrader. De må også telle antall kollisjoner. Til slutt skal du trekke konklusjon fra funnene.

2 Teori

For at hashmap implementasjonene skal være effektive bør de ha en god hashfunksjon. En hashfunksjon tar inn en verdi og omformer den til å bli en tallverdi slik at den kan brukes som index til en tabell.

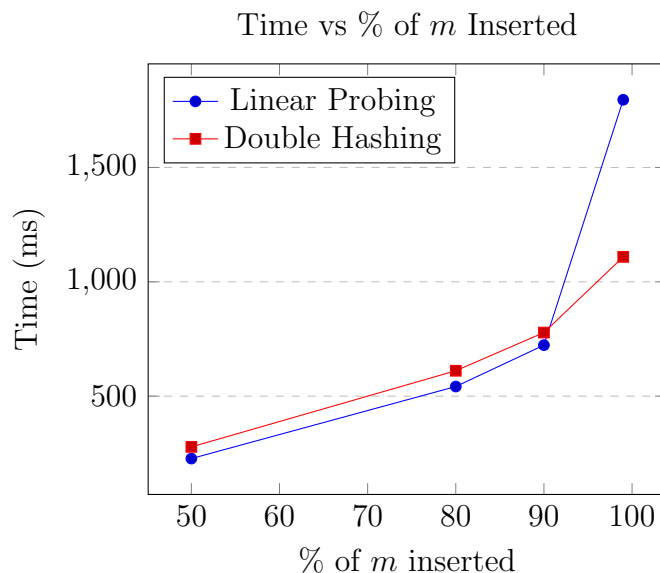
For deloppgave to gjør jeg dette ved å multiplisere verdien med et primtal og tar så modulus av størrelsen til tabellen. Tabellstørrelsen bør helst være et primtal. Jeg satt det til å være det nærmeste primtallet over 10 000 000 for å møte kravet, altså $m = 10\,000\,019$.

3 Resultater

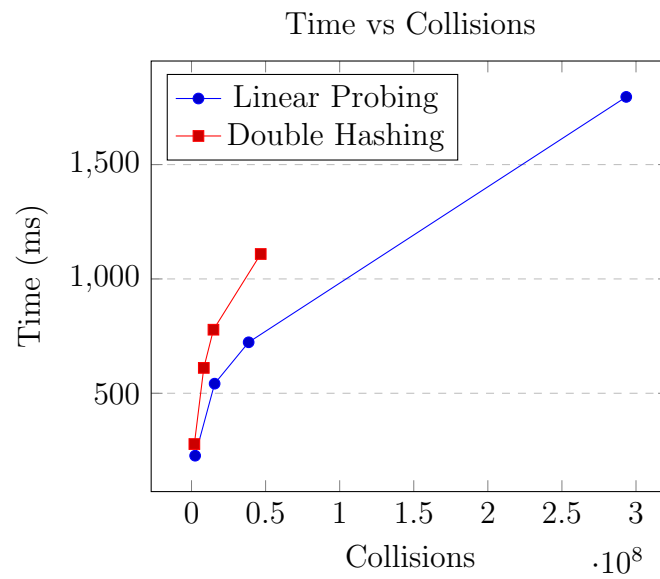
Etter omtrent ett minutt ender LinearProbing på omtrent $\text{index} = 9989405$ av $m = 10000019$. Dette er fordi hashfunksjonen er dårlig, og det er for mange kollisjoner. Vi måler derfor ikke tiden når den er 100

Tabell 1: Her ser du målingene for hashmap implementasjonene per fyllingsgrad.

Method	% of m inserted	Time (ms)	Collisions
Linear Probing	50	227	2,450,043
Linear Probing	80	542	15,601,418
Linear Probing	90	723	38,560,263
Linear Probing	99	1796	293,414,184
Double Hashing	50	278	1,925,374
Double Hashing	80	611	8,286,508
Double Hashing	90	778	14,800,364
Double Hashing	99	1109	46,712,940
Double Hashing	100	27613	788,629,257



Figur 1: Forholdet mellom tid og fyllingsgrad ser ut til å begynne lineært, men blir eksponentiell.



Figur 2: Forholdet mellom tid og antall kollisjoner ser ut til å være logaritmisk.

4 Konklusjon

Det tar betydelig mer tid for høyere fyllingsgrad og antall kollisjoner ser ut til å ha et omtrent logaritmisk forhold til tid. Dobbel hash er betydelig raskere enn lineær probing.