

Del 1

Det har blitt opprettet en hashtabell med lengde 127 (primtall) for å lagre studenters navn i algdatt høst 24 -klassen.

Når det legges til et navn i tabellen og det blir en kollisjon etter at navnet har blitt hashet, blir det håndtert med lenket liste. Det printes ut hver kollisjon når navn skal settes inn i tabellen samt tabellindeksen det ble kollidert på, som vist på bildet:

```
Collision during insertion: Edvard Granheim Harbo and Sigurd Spook at index 8  
Collision during insertion: Jon Martinus Rødtang and Eskild Smestu at index 61  
Collision during insertion: Helene Hauge Kjerstad and Garv Sood at index 66  
Collision during insertion: Jon Bergland and Håkon Sørli at index 81
```

Totalt antall kollisjoner fra vår hashfunksjon ble 37:

```
Collisions: 37  
Load factor: 0.97637796
```

Lastfaktoren er på omtrent 98%. Lastfaktoren er nærme 100%, noe som er bra fordi det betyr at gjennomsnittlig har hver indeks ett navn.

Det ble i tillegg gjort søk på gruppemedlemmenes navn samt et test navn som ikke var i hashtabellen. Hashfunksjonen og lenket listen klarte å finne igjen navnene som var i listen og ga false for testnavnet som ikke var i listen som ønsket:

```
HashTable contains Sander Sandvik Nessa: true  
HashTable contains Garv Sood: true  
HashTable contains Usman Ghafoorzai: true  
Collision during search: Ramtin Forouzandehjoo Samavat and negativeTest  
HashTable contains negativeTest: false
```

Antall kollisjoner per person er på omtrent 0.3:

```
Collisions per name: 0.2983871
```

Del 2

Resultatene for del 2 ble slik:

```
----- Del 2:-----  
  
Tid for innsetting i egen hashtabell: 529 ms  
Last faktor: 0.909091157024861  
Antall kollisjoner: 20942909  
Gjennomsnittlig antall kollisjoner per innsetting: 2.0942909  
Tid for innsetting i Javas HashMap: 1965 ms
```

Her viser den antall kollisjoner for innsetting i egen hashtabell, lastfaktoren, gjennomsnittlig antall kollisjoner per innsetting, samt tiden det tok for å sette inn 10 millioner tall i vår egen hashtabell kontra Java sin HashMap.

Vi kan se at lastfaktoren er nærme 1, noe som er bra. I tillegg kan vi se at én innsetting har gjennomsnittlig 2 kollisjoner.

Tiden for å sette disse tallene i vår egen Hashtabell tar mye kortere tid enn Java sin HashMap. Nesten $\frac{1}{4}$ så lang tid.