ICA 5

Kartlegge alle datamaskiner som gruppemedlemmene bruker og finne så

mange detaljer om deres bestanddeler / komponenter som mulig (CPU

frekvens, størrelse for RAM, L1, L2, osv.). Gi også en kort beskrivelse av

bestanddelene (hva er deres funksjon og hvordan de fungerer) samt lag en

tabell og presenter deres funn (ta også med detaljer om operativsystem).

CPU frekvens - *Hvor fort prosessoren jobber.*

L2 & L3 - *Minne i CPU. Veldig lavt minne, men utrolig raskt.*

RAM - *Har begrenset minne. Lagrer informasjon midlertidig så det raskt kan hentes opp.*

OS - *Operativsystem*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Terje** Macbook pro | **Erik** Lenovo | **Eirik** Lenovo | **Mathias** Macbook Air | **Johanne** Macbook Air | **Stina** Macbook Pro) |
| CPU frekvens | IntelCore i5  2,7GHz  Dual Core | A10-7300 with Radeon™ R6 Graphics 1.90GHz  Quad core | IntelCore i7 4720HQ - 2,7GHz  Quad Core | Intel Core i7 1,7 GHz  Dual Core | Intel Core i5  1,6 GHz  Dual Core | Intel Core i5  2,3 GHz  Dual Core |
| RAM | 8GB | 8GB | 16GB | 8GB | 4GB | 4GB |
| L2 | 256KB (?) | 4MB | 6MB | 256 KB | 256KB | 256KB |
| L3 | 3MB (?) |  |  | 4 MB | 3MB | 3MB |
| OS | OS X  10.11.3 (15D21) | Windows 10 | Windows 10 | OS X 10.11.3(15D21) | OS X  10.11.3 (15D21) | OS X  10.11.3 (15D21) |

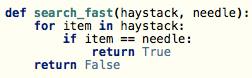
Søk for å finne tall 150128 av 1000000.

| Ytelse | Slow Search | Fast Search |
| --- | --- | --- |
| Terje | 0.623 | 0.088 |
| Stina | 0,756 | 0,265 |
| Johanne | 0,780 | 0,573 |
| Erik | 2.666 | 0.402 |
| Eirik | 0.546 | 0.079 |
| Mathias | 0.651 | 0.111 |

Søk etter ordet "denmark" i Hamlet.txt fra uke 06

| Ytelse | Slow Search | Fast Search |
| --- | --- | --- |
| Terje | 0.021 | 0.016 |
| Erik | 0.082 | 0.065 |
| Eirik | 0.017 | 0.018 |
| Stina | 0,079 | 0,054 |
| Johanne | 0,029 | 0.016 |
| Mathias | 0.028 | 0.014 |

Algoritmen



Algoritmen vi har brukt er O(N) algoritme innenfor BIG ’O algoritmene. O(N) algoritmen er lineær, noe som tilsier at en vekst i datagruppen (i dette tilfellet haystack) vil lede til en lik proporsjonal økning i søketiden, ettersom tiden det tar å søke gjennom hvert element er den samme. Eneste forskjellen mellom seach slow og search fast er at search slow må gå gjennom hele listen før den returnere svar.   
Hvis needle er lengre fremme i listen vil algoritmen returene fortere.

Link til Mathplotlib python prosjekt basert på tabellene ovenfor: <https://github.com/Mathih13/IS-105_2016_Gruppe-5/releases/tag/uke5>