

# Programering Lab 1

---

## 1dt909 Parallellprogrammering

Emil Ulvagården

### Run the program

```
go run BinaryHeap.go  
  
go run LinkedList.go  
  
go run md5Crack.go
```

### Uppgift 2

Låsningen när PushHead görs är så minimal som den kan bli då endast två noder låses och dessa är head och den dåvarande nästa noden. Den nya noden placeras emellan dessa noder och sedan låses de två låsta noderna upp. I PopHead låses head och den noden som ska plockas bort. Så länge som nästa node inte är tail sätts heads nästa till den node som ligger efter noden som ska bort. Vid PushTail låses först head som pred och sen låses noden efter head. Dessa lås flyttas genom listan tills pred har hamnat på den näst sista noden och det andra låset är på tail. Här sätts en ny node in och låsen låses upp. Poptail fungerar på liknande sätt som PushTail låses istället de två sista noderna som inte är tail. Detta för att sedan sätta den näst sista nodens nästa till tail. I contains fungerar låsningen på samma sätt som PushTail och PopTail. För PrintQueue låses hela kön tills hela kön har gått igenom.

### Uppgift 3

För generatePasswords används både en WaitGroup och kanaler. WaitGroupen används för att stänga kanalerna som skrivs till när alla möjliga lösenordskombinationer har skapats. Om lösenordet istället hittas av någon worker skrivs lösenordet till en kanal som markerar att lösenordet är hittat och programmet avslutas. Antalet go routines som kör generatePasswords är samma som antalet möjliga karaktärer för att försöka sprida ut vilka lösenord som kollas relativt jämt över alla olika kombinationer.

Workers	Time
1	2m 23s
2	2m 37s
4	3m 17s
8	3m 50s
256	4m 02s

Den tid som det tar för att hitta lösenordet ökar troligast med antalet workers pga att worker funktionen är mycket snabbare än generatePasswords som worker är beroende av för att kolla om det är rätt lösenord.