

## Övning 3 i Datastrukturer

Listor, listor och listor. Länkad lagring är grunden för det mesta vi gör i kursen.

Inga övningar behöver redovisas men tänk på att jag gillar att knyta an till övningsuppgifter och labuppgifter på tentan så det kan löna sig att ha gjort (och förstått) dessa.

Jobba gärna i grupp men se till så att alla "hinner med". Det är helt meningslöst att en löser uppgifterna och alla andra hummar instämmande. Det är mycket bättre om alla funderar en stund över en uppgift och att ni sedan diskuterar varför ni tänker som ni gör. Är ni ensamma så diskutera med er själva :-)

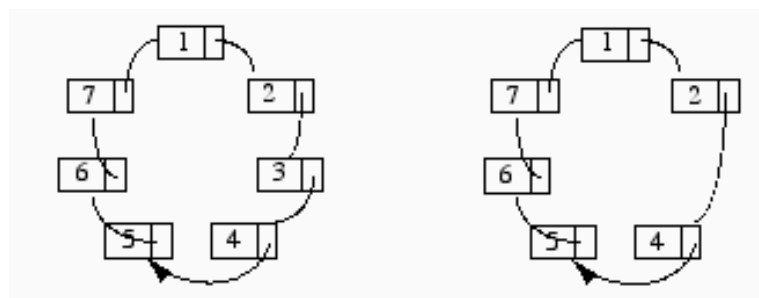
*Tänk på att man inte kan säga nåt om komplexitet utan att säga nåt om representation.*

1. Vilken komplexitet har operationen att sätta in ett element *sist* i en enkellänkad lista som består av  $n$  element?
2. Vilken komplexitet har operationen att sätta in ett element *först* i en enkellänkad lista som består av  $n$  element?
3. Vilken konkret representation för listor skulle du välja om det inte får finnas någon begränsning av längden på dem och det måste gå snabbt att lägga till ett nytt element både i början och i slutet på listan. Dessutom ska operationerna att skapa en tom lista och ta bort första elementet finnas.
4. En klassisk uppgift att öva länkad lagring på: Ett antal barn står i en cirkel och håller på med utplockningsrutinen "Ole dole doff". Ett av barnen som inser att sysselsättningen kan generaliseras, går hem till sin mamma som läst kursen i programmering och ber henne skriva ett program som gör följande:

Läser in två heltal 'mte' och 'antal' och därefter plockar bort var 'mte' person bland de 'antal' personerna som står i en cirkel i ordningen 1,2,3,...'antal'. Cirkeln krymper så fort som en person försvinner. Numren på de utplockade personerna skall successivt skrivas ut.

Exempel: Om antalet personer är 7 så är utgångscirkeln 1,2,3,4,5,6,7 och om 'mte' är 3 så skall talen 3,6,2,7,5,1,4 skrivas ut.

"Cirkeln" skall simuleras med en cirkulär länkad lista. När en person försvinner skall en nod tas bort.



Du skall skriva två program: ett som använder en enkellänkad lista och ett som använder en dubbellänkad och du skall inte använda listklasser från Java utan skall manipulera pekarna själv.