

En abstrakt datastruktur beskriver vilka operationer som finns men inte vad innehållet är, dvs. den beskriver inte lagrings sättet.

Datastrukturer

- Arrays
- Länkade listor
- Träd

Datalogiska klasser och datastrukturer (känt som abstrakta datatyper (ADT)) för denna kurs: kö, deque, stack, prioritetkö, binära träd.

Stack är precis som det låter, en trave med något. Detta betyder att varje gång du lägger på något nytt på stacken, så kommer du endast åt det du la på senast, detta förkortas ner till LIFO (Last in, First out). Tänk ett torn av tallrikar.

Stack-interfacet: <https://docs.oracle.com/javase/7/docs/api/java/util/Stack.html>

En kö fungerar som i verkligheten. Det första som tar sig in i kön, är även det första som lämnar kön, kallas även FIFO (First in, first out).

Kö-interfacet: <https://docs.oracle.com/javase/7/docs/api/java/util/Queue.html>

- add och offer metoden kallas för enqueue, dvs. förläng kön.
- poll och remove kallas för dequeue, dvs. förkorta kön.

En deque (double-ended queue) fungerar som en kö, dock med tillägget att det går att plocka och lägga till både framifrån och bakifrån.

I en prioritetkö plockas de värde med högst prioritet bort först. Så om du lägger in värdena 1 3 2 i den ordning, och sedan hämtar ett värde från kön så hamnar 3:an först. Detta kallas även max prioritetkö.

Länkade listor

Ibland så kan det vara väldigt opraktiskt att lagra datan i form av en array. Arrays är exempelvis indexerade, vilket betyder att om du exempelvis vill lägga in ett värde första i arrayen så behöver du flytta alla andra värden ett steg för att behålla data.

I en länkad lista finns endast noder(nod = data med en pekare), noder som vet om vad som kommer näst på tur men inte mer än så. Detta gör så att om du behöver plocka bort en nod ur listan så stör inte det resten av listan, eftersom alla förutom (högst) en nod, dvs. den som låg före noden som tas bort, visste om att den borttagna noden var där.

Träd

Träd är som länkade listor bara det att varje nod kan ha flera noder, då kallas de för barn, alternativt löv. Följande är träd-begrepps

Noder är de objekt som trädet är uppbyggt av. De innehåller data och pekare.

Rot är den översta noden i trädet. Den pekas inte ut av någon annan nod.

Barn till en nod är de som pekats ut av noden.

Förälder är noden ovanför i trädet.

Syskon har samma förälder.

Löv är en nod vars pekare är None.

Delträd definieras så här: En godtycklig nod i trädet kan ses som en rot, och den , tillsammans med alla noder under den (barn, barnbarn osv.) bildar ett delträd.

Nivå är det antal steg från roten noden befinner sig. Roten är på nivå noll.

Höjd är den maximala nivån som någon av trädets noder befinner sig på.

Balanserat är binärträdet om skillnaden i höjd mellan höger och vänster delträd till varje nod är noll eller ett.

Fullt är binärträdet om alla noder utom löven har exakt två barn, och alla löv är på samma nivå

Binärt Träd

Ett binärt träd är ett träd vars alla barn endast har två pekare. Det finns även en variant av binärt träd som kallas binärt sök träd, där placeras alla värden lägre än roten till vänster, och de högre till höger.