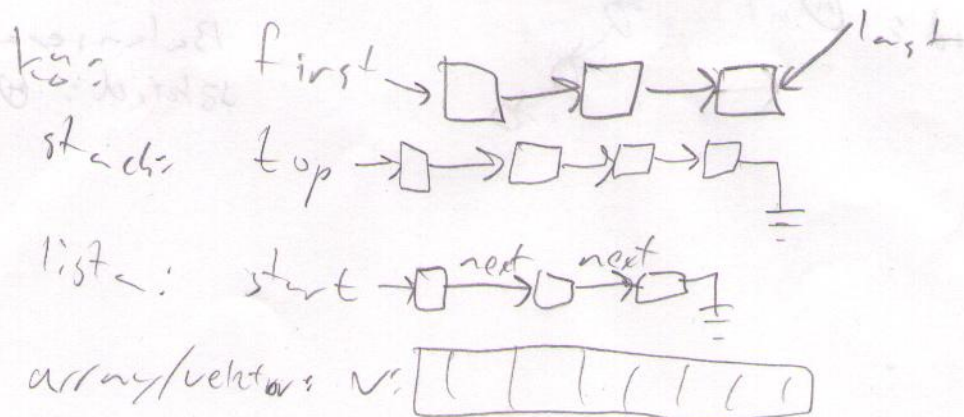


Datastrukturer

- Längd
- Änd
- Prioritetskö
- Förberäknad-funktion
- Trä
- Hash tabeller
- Lattmann hashing
- Skiplistor

Översikt över kända datastrukturer

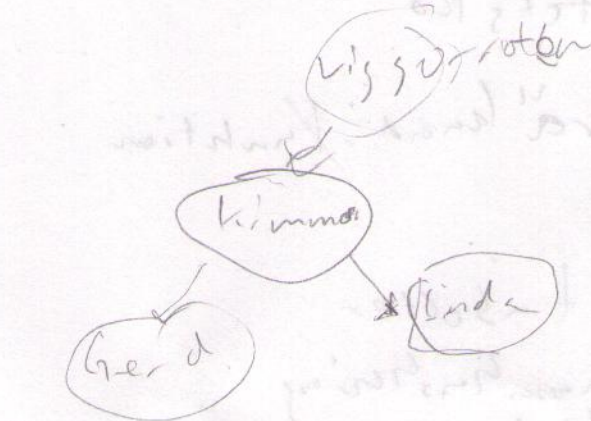
Linjära strukturer:



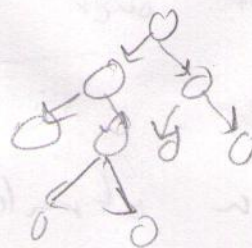
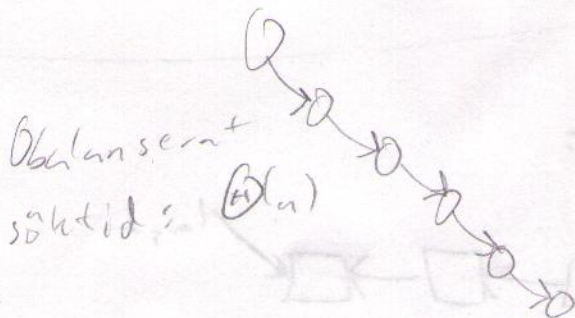
Binära sökträd

ett träd som är sorterat så att mindre element är till vänster kallas sökträd.

Exempel:



n st



Balanserat
söktid: $\Theta(\log n)$

Balanserade träd

Problem med binära sökträd:

Balanserade träd tar tid

Röd svart träd (inte jätte snabba)

Sökning

Använd vanligt träd sökningsalgoritmen
som tar tid O

Prioriterade köer

• Datastruktur där varje post har
en prioritet, det vill säga ett tal
som anger hur viktig posten är.

Iterpar — snabb representation av
prioritetsköer

Komplett binärträd —

alla noder i trädet är fyllda
utom de sista.

Realisering av operationen Remove i en heap

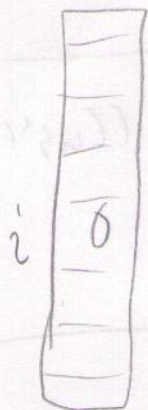
Elementet som ska tas bort

Reorganiserad funktion

The retrieval

Bn, brie är en implementation av en
mängd strängar som ett träd.

$f(x)$ ger index



The standard reference for this is
Knuth's "The Art of Computer Programming and
Searching", chapter 6.4. He recommends
the hash

```
for (hash=0; hash<len; hash++)  
{  
    hash = (hash * 256 + str[hash]) % mod;  
}
```

• The keys are unaligned - variable length byte arrays.

• Sometimes keys are several such arrays.

• Sometimes a set of independent hash functions were required.

#define hashsize(n) ((n+1) << 1)

n

/* handle most of the key
while (len >= 12)

Latnanna hushadn

Buska bara på dem tre första bokstäverna
i söknnyckeln. Använd sedan binär sökning.

1, 1, 1

Skipplistor

- Probabilistiska datastrukturer
- Enkel att implementera
- I allmänhet lika snabb som balanserade träd att söka i och enklare att ändra på.

1, 1, 1

Sökning i skipplistor

1, 1, 1

Analys av skippistor

skippistor i praktiken