4.5 Indeksien toteuttaminen

- standarditapa: käänteistiedosto
- muita: puurakenteet .. → suffiksitaulukot
- hajautus → nimikirjoitustiedostot

1° Käänteistiedosto
(vrt. luku 2)

sanasto (vocabulary) = termien luettelo
+ esiintymälistat (sanoihin, merkkeihin)

Käänteistiedoston tilantarve?

- sanasto: $O(n^{\beta})$, missä β välillä 0..1, yleensä 0.4 .. 0.6.

Esim. TREC-2 kokoelma (1 GB), sanasto 5 MB.

(Heaps'in laki: $V = Kn^{\beta}$, $K \sim 10 .. 100$)

- esiintymälistat: O(n)
hukkasanojen poistolla käytännössä
lisätilan tarve n. 0.4n

Eri vaihtoehtoja:
- full inverted: sanoittain (tai jopa merkeittäin)
- lohkoittain:
osoittimet pienemmiksi
lohkon monille esiintymille vain yksi osoitin

Esim. (BYRN, s. 193 ..)

This is a text. A text has many words. Words are made from letters.

sanasto	esiintymät	(osoitin merkkipositioon)
letters	60	
made	50	
many	28	
text	11, 19	
words	33, 40	

- lohkoittain:

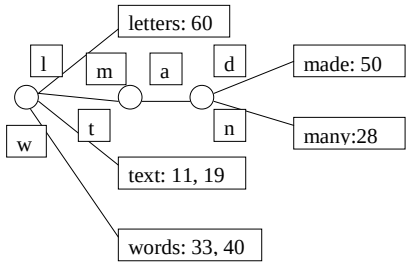
This is a text.	A text has many	words. Words are	made from letters.
1	2	3	4

letters	4
made	4
many	2
text	1, 2
words	3

- lisätilantarve jopa vain n. 5 %
- lähekkäisyyskyselyissä tarvitaan lohkojen selaus
(tekstin tulee olla saatavissa)
- lohkojen määrä esim. 256 tai 64KB (osoitin 1 tai 2 tavua)
(sanaosoittimet esim. 4 tavua)
- lohkot voivat olla erikokoisia (esim. lauseita, sivuja)

Käänteistiedoston sanasto voidaan organisoida eri tavoin:
- järjestetty tiedosto: binäärihaku
- hajautus: haun kustannus ei riipu tekstin koosta
- trie (merkkipuu)

Muodostus trie-rakenteena on yksinkertaista (ei vaadi varsinaisesti järjestämistä).



- indeksi jaetaan usein kahteen tiedostoon:
esiintymät peräkkäin
sanasto toiseen tiedostoon järjestyksessä,
jokaiselle sanalle osoitin vastaavan
esiintymälistan alkuun
- sanasto ehkä mahtuu keskusmuistiin,
esiintymien lkm selviää helposti
- sanaston sivutus on tehotonta; BYRN esittää
osittamisratkaisun

2° Suffiksipuut ja –taulukot

- sanaperustainen käsittely ei aina riitä (fraasit)
tai ei ole muuten käypä (merkkijonotieto esim. genetiikassa)

Suffiksipuu: sekä sanaperustainen että yleistetty käsittely,
puun rakentaminen työläämpää

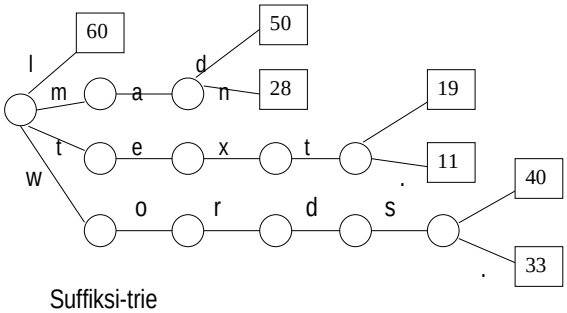
Periaatteet:
- tarkastellaan tekstiä yhtenä pitkänä merkkijonona
- jokainen merkki(positio) vastaa yhtä tekstin loppuosaa eli
suffiksia (merkistä koko tekstin loppuun)
(yksikäsitteisesti, lopetusmerkki tekstin lopussa)
- loppuosat muodostetaan vain sanojen aluille (tai muille
valituille indeksipositioille)

Esimerkki:

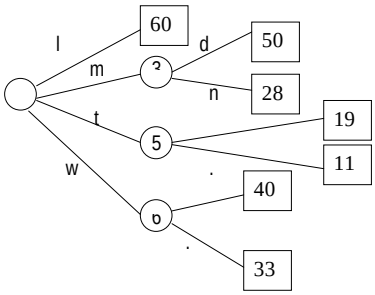
This is a text. A text has many words. Words are made from letters.

text. A text has many words. Words are made from letters.
text has many words. Words are made from letters.
many words. Words are made from letters.
words. Words are made from letters.
Words are made from letters.
made from letters. (suffiksit)
letters.

Suffiksit tallennetaan trie-rakenteeseen: haku suffiksin alkuosalla, itse suffiksi lehtisolmussa.



Suffiksi-trie

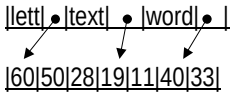


Suffiksipuu (tiivistetty Patricia-puun tapaan)

Suffiksipuu vie aika paljon tilaa. Suffiksi-informaatio voidaan kuitenkin pakata tehokkaasti taulukkoon:

|60|50|28|19|11|40|33|

- kaikki osoittimet järjestyksessä, binäärihaku mahdollinen –
- tiedoston tulee olla saatavilla, levyhaut voivat hajaantua laajalle alueelle (eri lohkoihin) eli yleisessä tapauksessa tehoton
- voidaan rakentaa 'päällysindeksi' muutamien suffiksien alkuosista.



Päällysindeksi voi olla pieni, voidaan tehdä monella tavalla.

3° Nimikirjoitustiedosto (signature file)

- hajautukseen perustuva indeksirakenne:
  - · teksti jaetaan lohkoihin
  - · lohkolle lasketaan bittipeite yhdistämällä sen sanojen hajautusarvot (OR-operaatiolla)
  - · hakusanaa tutkimalla tiedetään, voiko sana esiintyä lohkossa:
    - lasketaan hakusanan hajautusarvo h'
    - jos h':n bitit esiintyvät lohkon hajautusarvossa, sana voi esiintyä lohkossa
    - lohkon hajautusarvossa mainitut bitit voivat olla mukana myös joidenkin muiden sanojen esiintymien takia (false drop) → sanan todellinen esiintyminen on tarkistettava erikseen

Esimerkki.

| This is a text.| A text has many | words. Words are | made from letters.|
1 2 3 4

- lohkojen nimikirjoitukset:
  000101 110101 100100 101101

h(text) = 000101
h(many) = 110000
h(words) = 100100
h(made) = 001100
h(letters) = 100001

- olkoon h(test1) = 101010
  'test1' ei esiinny missään lohkossa
  (eikä sitä tarvitse tarkistaa)
  h(test2) = 100101
  'test2' näyttää kuuluvan sekä lohkoon 2 että 4
  (kumpikin paljastuu 'false drop'iksi)

False drop –todennäköisyys on käänteisessä suhteessa rakenteelle varattuun tilaan (nimikirjoitusten pituuteen).

Esim. 10 %:n lisätilaa käytettäessä tn = 2 %,
20 %:n lisätilaa käytettäessä vain 0.046 %

- fraasien etsiminen ja lähekkäisyyskyselyt sopivat (lohkon sisällä)
- lohkot voidaan määritellä päällekkäisiksi

- tehokkuus ei riitä suurille tiedostoille (ei liene nykyisin yleisessä käytössä)