Søketek uke 3

Gruppe 1

Agenda

- Første time
  - Assignment b-1
  - Repetisjon
  - Ukas shoutout
- Andre time
  - Selvstendig jobbing

Assignment b-1

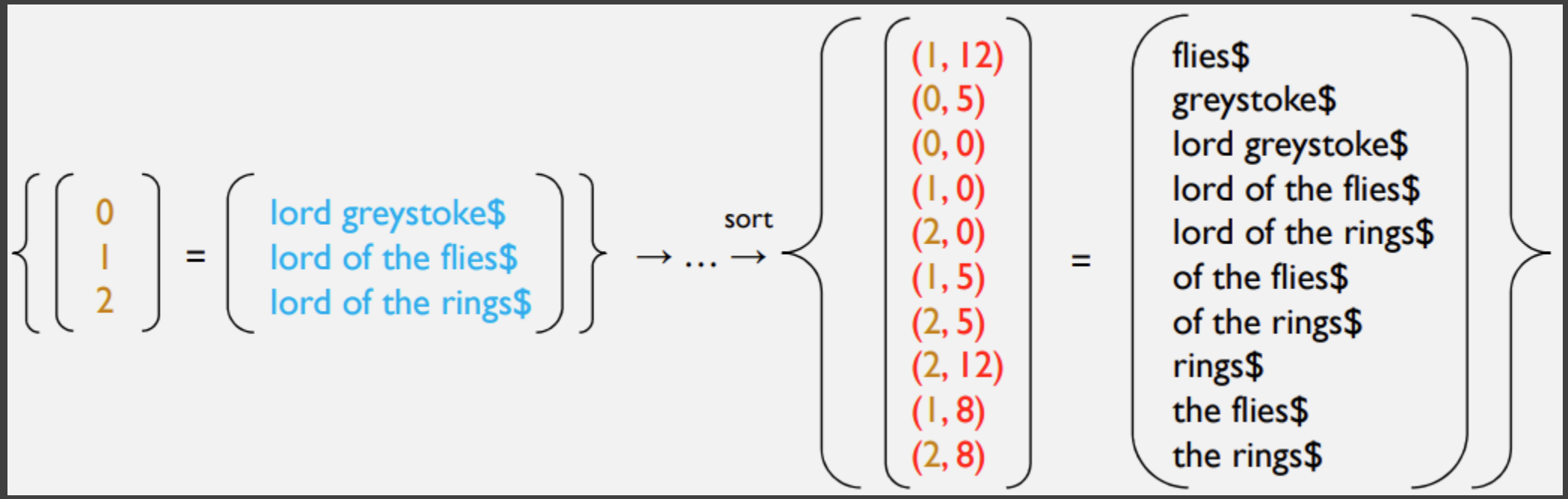
- suffixarray
  - Suffix array
  - Binære trær/Binærsøk
- stringfinder
  - Tries
  - Aho-Corasick

Suffix array

- Hvis vi vil finne matcher vilkårlig langt inne i en tekst
  - «Et prefix av et suffix er et infix»

- Finne alle suffixer i dokumenter
  - lagre som (docId, offset)
- Sortere leksikografisk
  - Til obligen: sjekk ut key for sortering
- Kan utføre binærsøk

Suffix array



Nåla i høystakken

```
(0, {"field1": "Japanese リンク", "field2": "HallO HALlo"})
(1, {"field1": "en LiTeN test", "field2": "sØketek er GØY"})

self.__haystack = [
    (0, “japanese リンク \x00 hallo hallo"),
    (1, "en liten test \x00 søketek er gøy")
]
```

Nåla i høystakken

```
self.__haystack = [
    (0, “japanese リンク \x00 hallo hallo")
]
```

self.__suffixes skal inneholde (docID, offset)-par

```
self.__suffixes = [(0, 0), (0, 9), (0, 15), (0, 21)]
```

Og sorteres leksikografisk

```
self.__suffixes = [(0, 21), (0, 15), (0, 0), (0, 9)]
```

Suffix array utdrag

- Eksempel på utdrag fra et suffix array i oblig b-1

[1002, 505], [738, 522], [915, 0], [694, 177], [497, 199], [539, 1082], [720, 1634], [124, 549], [84, 210], [346, 495], [93, 2697], [610, 303], [1334, 803], [79, 866], [1007, 281], [13, 514], [362, 1316], [81, 1765]

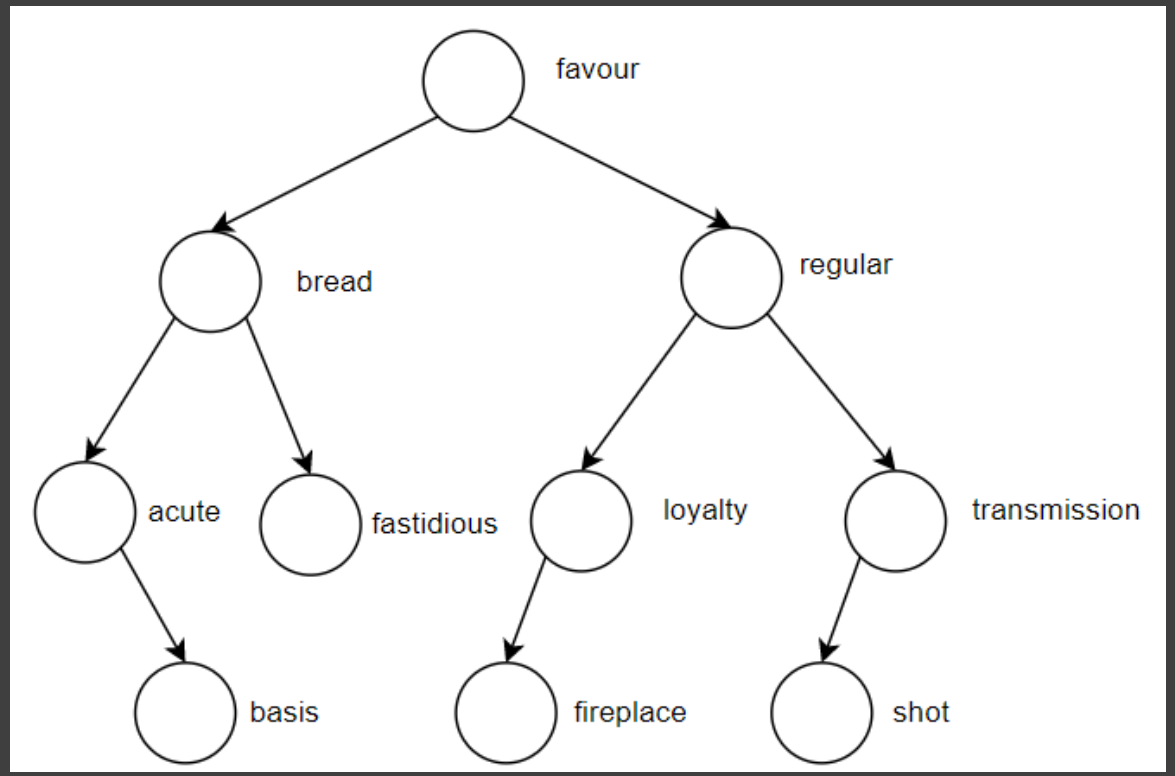
Binærsøk

- Samme prinsipp som binære trær
- Halvere søkerommet for hver iterasjon
  - Hoppe til høyre hvis større, til venstre hvis mindre

Binærsøk - eksempel

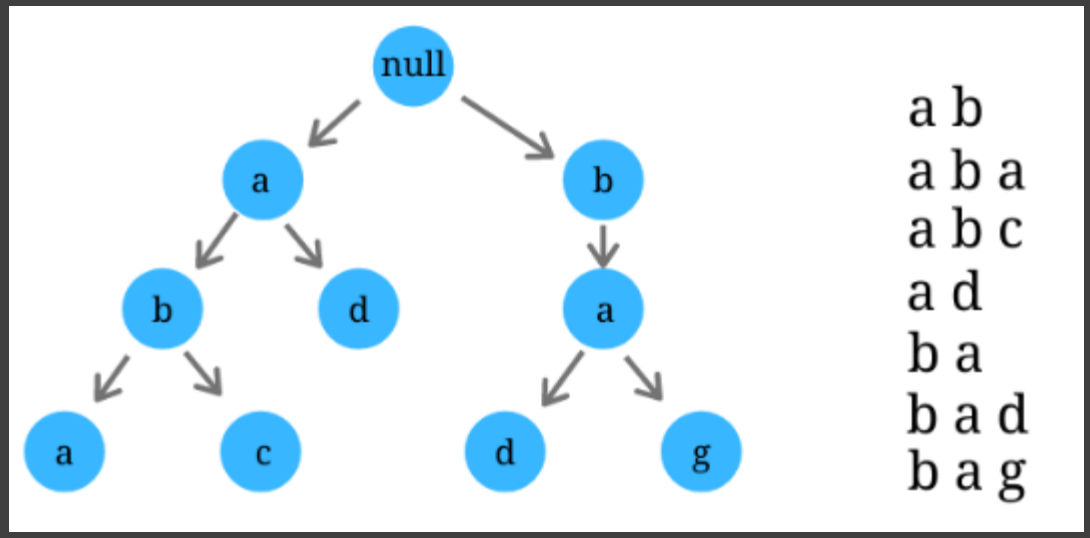
```
["acute", "basis", "bread", "fastidious", "favour", "fireplace", "loyalty", "regular", "shot", "transmission"]
```

- Eksempel: søke etter «ABBA»
- Vi beveger oss til venstre 2 ganger og returnerer
- Søkerommet halvert for hvert hopp, trengte ikke sjekke hele lista



Tries

- Tre-struktur med noder som bokstaver
- Ta med bokstaven på veien



Aho-Corasick (i obligen)

- Utfører en trie-walk (vi går gjennom en trie og ser etter matcher)
- Ha en liste med live states
  - Oversikt over hvor i trie-en vi er når vi traverserer
- Vi har funnet en match når vi treffer en final node
- Verdt å sjekk ut i trie.py:

```
def consume(self, prefix: str) -> Optional[Trie]:
def is_final(self) -> bool:
```

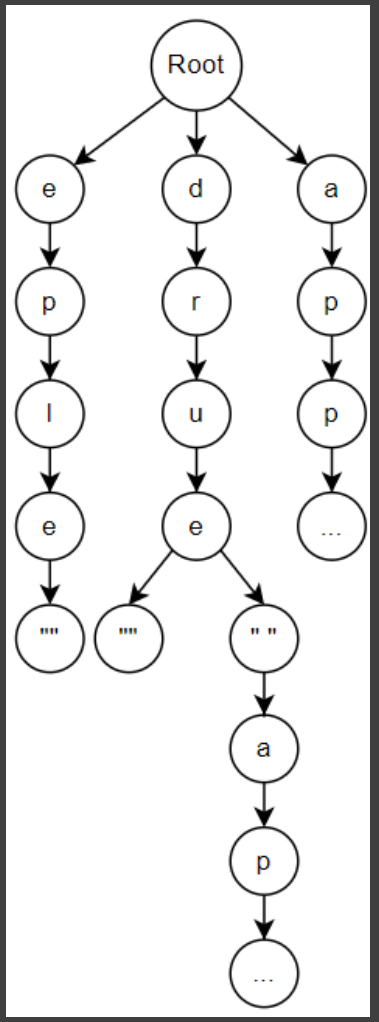
- Video: https://www.youtube.com/watch?v=O7_w001f58c&ab_channel=NiemaMoshiri

Traversering i en trie

```
def test_scan_matches_and_spans(self):
    strings = ["eple", "drue", "appelsin", "drue appelsin rosin banan papaya"]
    trie = in3120.Trie.from_strings(strings, self.__normalizer, self.__tokenizer)
    finder = in3120.StringFinder(trie, self.__normalizer, self.__tokenizer)
    results = list(finder.scan("et EPLE og en drue   appelsin  rosin banan papaya frukt"))
```

- Når vi scanner for «appelsin» ønsker vi både resultater fra roten og fra en state som har traversert «drue»

```
eple
drue
appelsin
drue appelsin rosin banan papaya
```



Agenda

- Første time
  - Assignment b-1
  - Repetisjon
  - Ukas shoutout
- Andre time
  - Selvstendig jobbing

Repetisjon

- Wildcards
- Permuterm index
- K-gram index
- Edit distance (Damerau Levenshtein)
- B-trær

Wildcards

- Spørringer som kan gi ulike svar
- Fi* → alt som starter på «Fi»
- Fi*er → alt som starter på «Fi» og slutter på «er»
  - Fishmonger, Filibuster, …
- Kan bruke permuterm-index eller k-gram index for søk

Permuterm index

- Index med rotasjoner av termer
- Transformerer wildcard-søk til prefix-søk
- Kan bruke resultatet i en invertert indeks

Permuterm index

- Må få * på slutten (prefix-søk når vi vet hva starten er)
- **i*ks$ → ks$i***
  - Matcher **ks$inde**

```
permuterm_index = {
    indeks$: "indeks",
    ndeks$i: "indeks",
    ...
    ks$inde: "indeks",
    s$indek: "indeks"
}
```

K-gram index

- Bigram index hvis lengde = 2
- Lage alle sekvenser av lengde k i hver term
  - Indeks → **i, in, nd, de, ek, ks, k**
- Ha en ekstra invertert indeks som fra k-grams til termer i ordboka

K-gram index

- Indeks → **i, in, nd, de, ek, ks, k**
- Query: **i*ks$**
- Intersect **i, ks, s** og søk med resultatet

```
bigram_index = {
    $i: ["indeks", "ingen", "instans"],
    ...
    nd: ["and", "indeks", "ondskap"],
    ...
    ks: ["indeks", "laks", "øks"],
    s$: ["is", "indeks", "instans"]
}
```

Edit distance

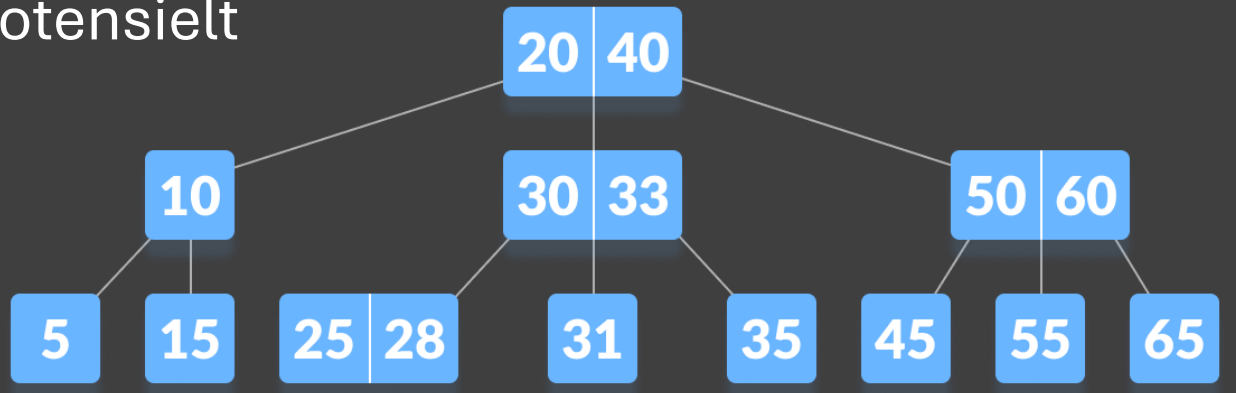
- Minste antall edits for å skrive om en streng til en annen
- 4 edit-operasjoner: Insert, delete, replace, transpose

		E	L	E	P	H	A	N	T
	0	1	2	3	4	5	6	7	8
R	1	1	2	3	4	5	6	7	8
E	2	1	2	2	3	4	5	6	7
L	3	2	1	2	3	4	5	6	7
E	4	3	2	1	2	3	4	5	6
V	5	4	3	2	2	3	4	5	6
A	6	5	4	3	3	3	3	4	5
N	7	6	5	4	4	4	4	3	4
T	8	7	6	5	5	5	5	4	3

- Video (uten transpose): https://www.youtube.com/watch?v=MiqoA-yF-0M&t=872s&ab_channel=BackToBackSWE

B-trær

- Binære trær: noder kan ha 2 barn (større og mindre enn seg selv)
- B-trær: noder kan ha forskjellig antall barn
- Kan dele opp i f.eks. større enn, mindre enn, mellom
  - Tradeoff: Kortere tre, men potensielt flere sammenligninger



- Video: https://www.youtube.com/watch?v=K1a2Bk8NrYQ&ab_channel=SpanningTree

Agenda

- Første time
  - Assignment b-1
  - Repetisjon
  - Ukas shoutout
- Andre time
  - Selvstendig jobbing

Ukas shoutout

- Jakob Schøien & Fridtjof Josefsen
- De har sykt mye bra
  - Flex
  - Kollektivet
  - Hit for hit

Agenda

- Første time
  - Assignment b-1
  - Repetisjon
  - Ukas shoutout
- Andre time
  - Selvstendig jobbing