Søketek uke 4

Gruppe 1

Agenda

- Tips og triks til stringfinder i obligen
  - Hvordan det funker
  - Hvorfor det funker
- Ukas shoutout
- Selvstendig jobbing resten av tiden

stringfinder

- NB: Ikke presis hvordan metoden implementeres, men intuisjonen bak
  - tar ikke hensyn til normalisering, metadata, etc.
- Hva skal vi egentlig gjøre?
  - Vi har et tekst-buffer og en ordbok/vokabular
  - Skal finne alle ordene/setningene som dukker opp både i bufferet og ordboka (finne matcher)
  - Og vi skal gjøre det fort!

3 ting å huske på

For hver term:

1. Kan jeg traversere fra roten?
2. Hvilke aktive states kan jeg traversere fra?
3. Lander jeg på en final node?

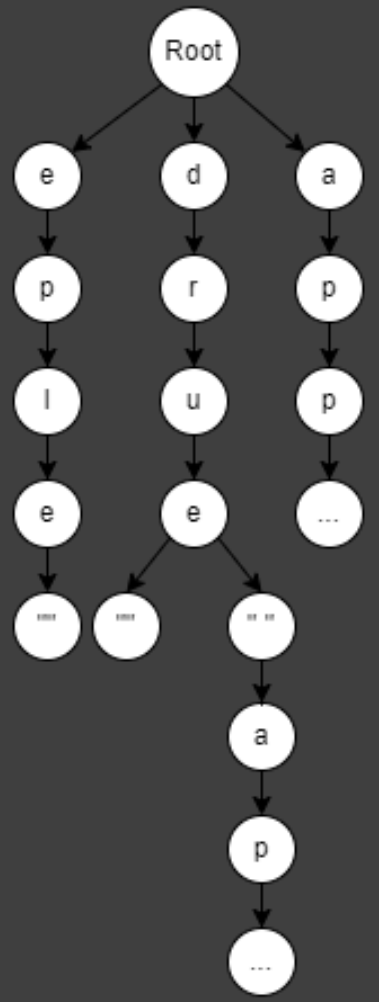
Eksempel

```
strings = ["eple", "drue", "appelsin", "drue appelsin rosin banan papaya"]
finder.scan("et EPLE og en drue   appelsin  rosin banan papaya frukt"))
```

Vi vil returnere `["eple", "drue", "appelsin", "drue appelsin rosin banan papaya"]`

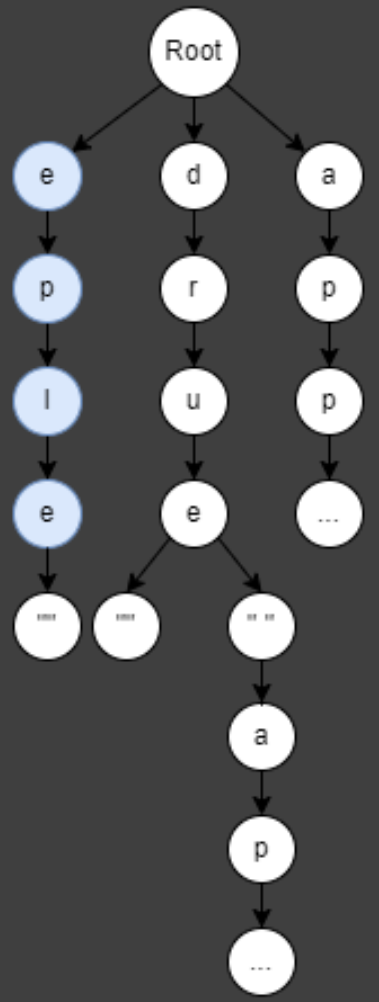
Eksempel

```
strings = ["eple", "drue", "appelsin",
           "drue appelsin rosin banan papaya"]

text = "et eple og en drue appelsin rosin banan papaya frukt"


states = [


]




matches = []
```



Eksempel

```
strings = ["eple", "drue", "appelsin",
           "drue appelsin rosin banan papaya"]

text = "et eple og en drue appelsin rosin banan papaya frukt"


states = [
        Root → "eple"
]
```

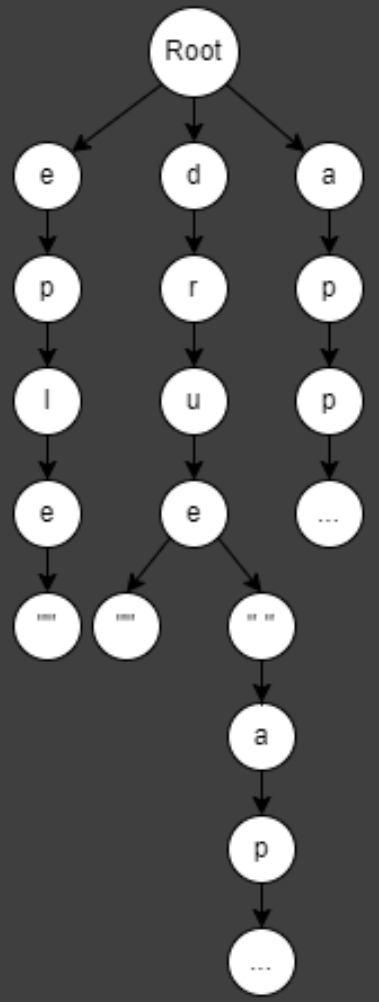



```
matches = ["eple"]
```

Eksempel

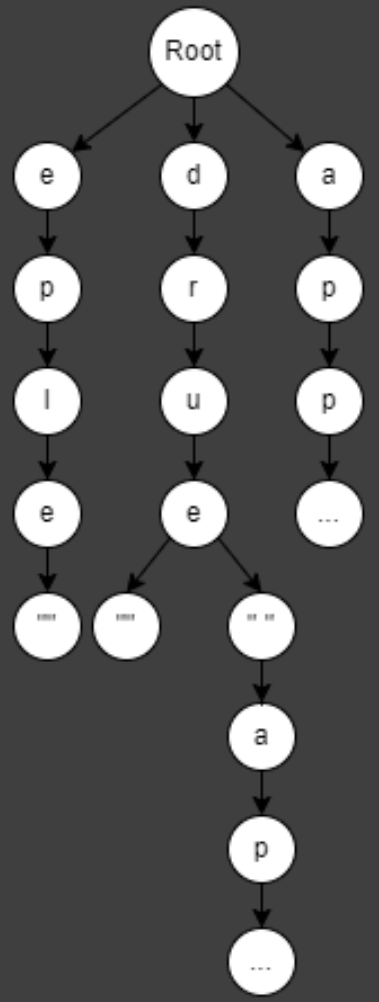
```
strings = ["eple", "drue", "appelsin",
           "drue appelsin rosin banan papaya"]

text = "et eple og en drue appelsin rosin banan papaya frukt"


states = [

        Root → "eple"
]
```

```
matches = ["eple"]
```



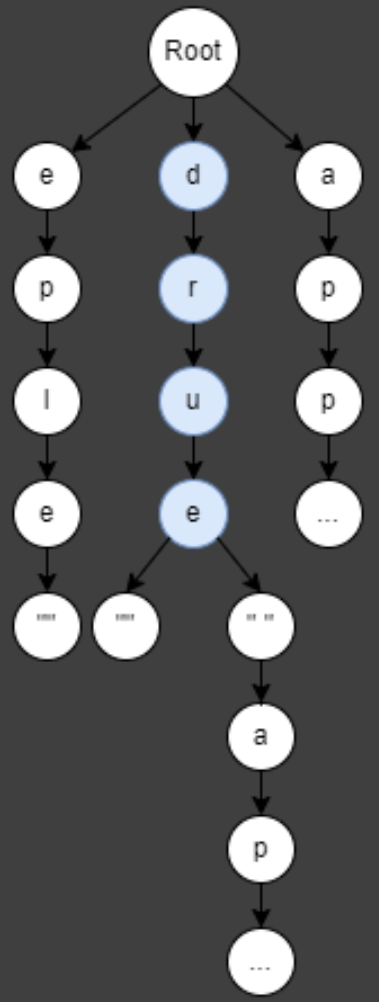
Eksempel

```
strings = ["eple", "drue", "appelsin",
           "drue appelsin rosin banan papaya"]

text = "et eple og en drue appelsin rosin banan papaya frukt"


states = [


]




matches = ["eple"]
```



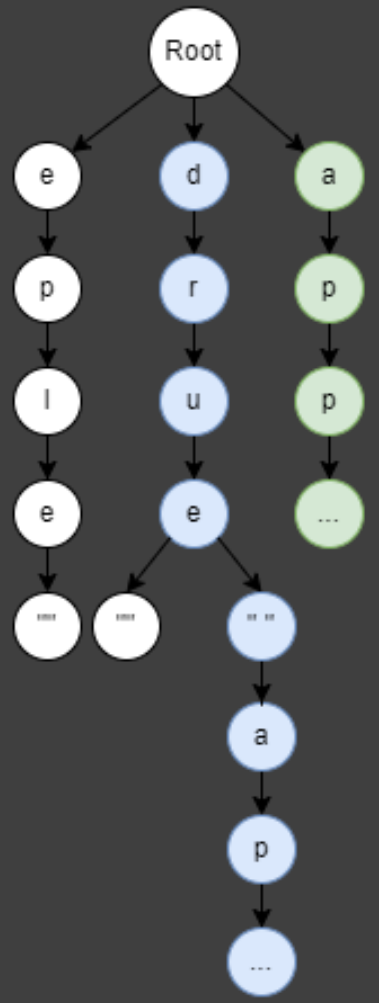
Eksempel

```
strings = ["eple", "drue", "appelsin",
           "drue appelsin rosin banan papaya"]
text = "et eple og en drue appelsin rosin banan papaya frukt"

states = [
        Root → "drue"
]

matches = ["eple", "drue"]
```



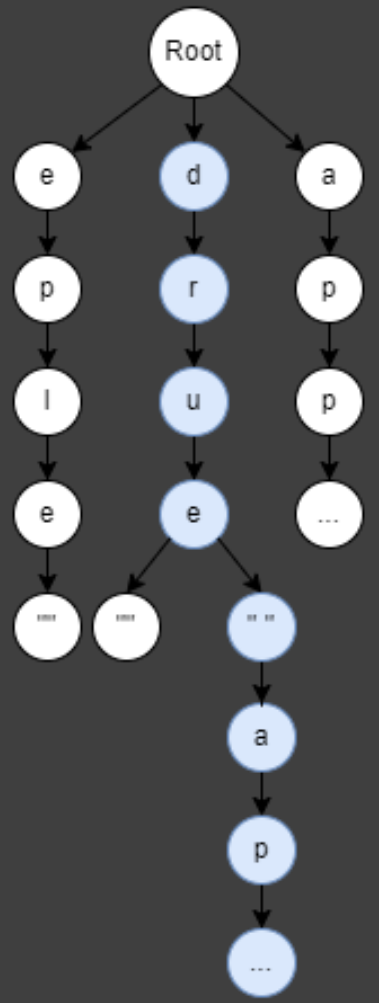
Eksempel

```
strings = ["eple", "drue", "appelsin",
           "drue appelsin rosin banan papaya"]
text = "et eple og en drue appelsin rosin banan papaya frukt"

states = [
        Root → "drue appelsin",
        Root → "appelsin"
]


matches = ["eple", "drue", "appelsin"]
```



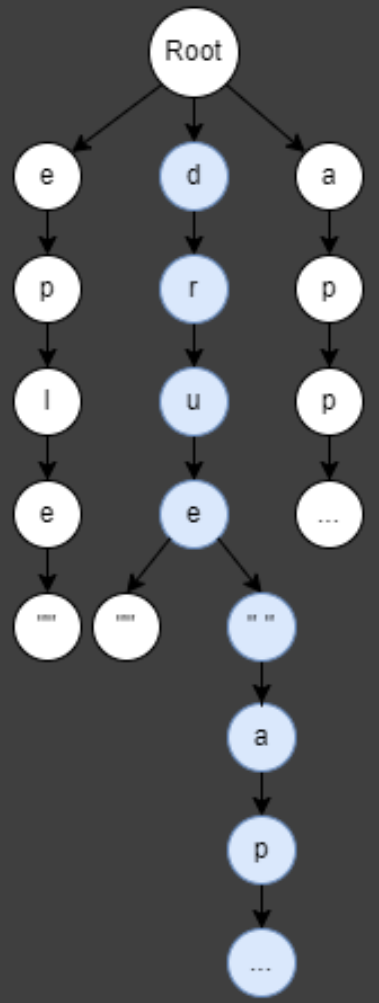
Eksempel

```
strings = ["eple", "drue", "appelsin",
           "drue appelsin rosin banan papaya"]
text = "et eple og en drue appelsin rosin banan papaya frukt"

states = [
        Root → "drue appelsin rosin",
        Root → "appelsin"
]

matches = ["eple", "drue", "appelsin"]
```



Eksempel

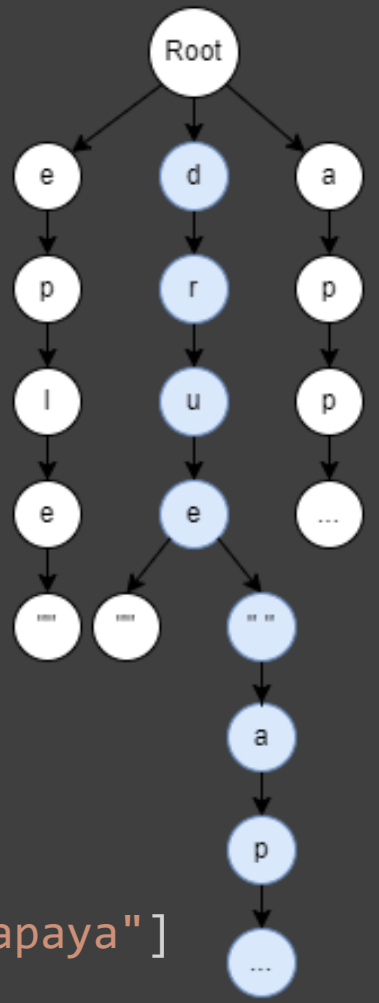
```
strings = ["eple", "drue", "appelsin",
           "drue appelsin rosin banan papaya"]
text = "et eple og en drue appelsin rosin banan papaya frukt"

states = [
        Root → "drue appelsin rosin banan"
]
```




```
matches = ["eple", "drue", "appelsin"]
```

Eksempel

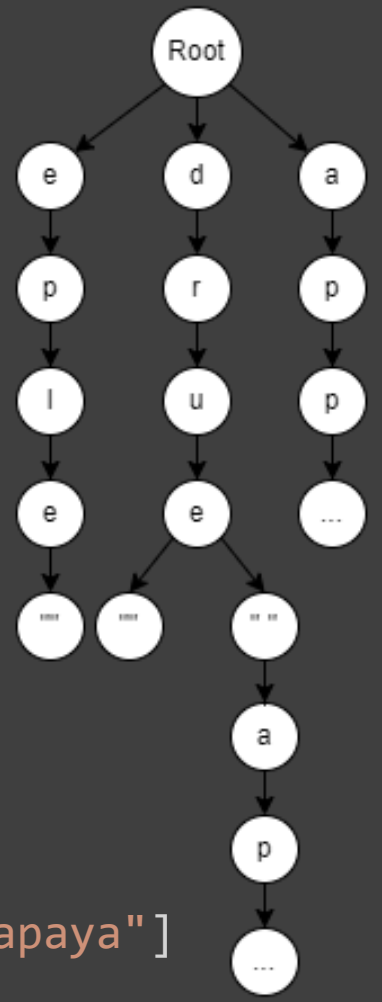
```
strings = ["eple", "drue", "appelsin",
           "drue appelsin rosin banan papaya"]
text = "et eple og en drue appelsin rosin banan papaya frukt"

states = [
        Root → "drue appelsin rosin banan papaya"
]
```




```
matches = ["eple", "drue", "appelsin", "drue appelsin rosin banan papaya"]
```

Eksempel

```
strings = ["eple", "drue", "appelsin",
           "drue appelsin rosin banan papaya"]

text = "et eple og en drue appelsin rosin banan papaya frukt"


states = [
        Root → "drue appelsin rosin banan papaya"
]
```




```
matches = ["eple", "drue", "appelsin", "drue appelsin rosin banan papaya"]
```

Agenda

- Tips og triks til stringfinder i obligen
  - Hvordan det funker
  - Hvorfor det funker
- Ukas shoutout
- Selvstendig jobbing resten av tiden

Hvorfor skalerer dette så bra?

- Fordi vi går gjennom terms 1 gang og det er sjelden mange aktive states.
- Hvorfor er det få aktive states?

Hvorfor er det få aktive states?

- States er en oversikt over «midlertidige matcher»
- Vi fjerner/pruner alle irrelevante states (det som ikke kan traverseres med nåværende term)
- Hva er en relevant state?
  - Oppretter ny state hvis vi kan traversere nåværende ord fra roten
  - Traversere eksisterende state hvis vi kan traværsere fra forrige posisjon
- Hvorfor er det trygt å slette så mange states?

2 regler for matching

1. Vi må traversere hele veien (i ett løp) for hver faktiske match
   - Hvis en state blir irrelevant, er det fordi «feil ord» dukker opp. Hvis ordene ikke kommer i presis riktig rekkefølge, vil de aldri kunne matche. Da vil dette forsøket være over, og vi kan starte på nytt senere. Neste forsøk må starte fra roten (ellers ville vi bare delvis matchet). Vi kan ikke «traversere litt» og fortsette senere
2. Det opprettes alltid en state for hver potenielle match
   - Nåværende term prøver å traversere alle states, inkludert fra roten. Hvis vi kan traversere fra roten blir denne staten alltid lagt til

Eksempel (1)

```
strings = ["eple", "drue", "appelsin", "drue appelsin rosin"]
"et EPLE og en drue appelsin appelsin rosin banan papaya frukt"
```

- Når «appelsin» dukker opp 2 ganger på rad vil vi *aldri* matche «drue appelsin rosin». Derfor trenger vi ikke sjekke om «rosin» passer etterpå, og kan slette staten.
- Hvis vi tar vare på staten 'Root → «drue appelsin»' i tilfelle «rosin» dukker opp senere, får vi ikke en komplett match
  - Ellers ville "drue appelsin dette matcher ikke rosin" gitt en match

Eksempel (2)

```
strings = ["drue appelsin rosin", "appelsin appelsin rosin"]
"drue appelsin appelsin rosin banan"
```

- Oppretter vi en state ‘Root → «drue»’ :

states = [

Root → «drue»

]

Eksempel (2)

```
strings = ["drue appelsin rosin", "appelsin appelsin rosin"]
"drue appelsin appelsin rosin banan"
```

- Lager vi en state: ‘Root → «appelsin»’ og fortsetter den eksisterende staten ‘Root → «drue appelsin»’

states = [

Root → «appelsin»,

Root → «drue appelsin»

]

Eksempel (2)

```
strings = ["drue appelsin rosin", "appelsin appelsin rosin"]
"drue appelsin appelsin rosin banan"
```

- Lager vi en state: ‘Root → «appelsin»’ og fortsetter den eksisterende staten ‘Root → «appelsin appelsin»’, og sletter den nå irrelevante staten ‘Root → «drue appelsin»’

states = [

Root → «appelsin appelsin»,

Root → «appelsin»

]

Hvorfor kan vi slette så mange states?

- Fordi (1) alle matcher alltid må traversere fra roten hele veien i presis riktig rekkefølge, og (2) alle relevante states alltid blir opprettet, kan vi slette alle states som blir irrelevante.

Agenda

- Tips og triks til stringfinder i obligen
  - Hvordan det funker
  - Hvorfor det funker
- Ukas shoutout
- Selvstendig jobbing resten av tiden

Ukas shoutout

Agenda

- Tips og triks til stringfinder i obligen
  - Hvordan det funker
  - Hvorfor det funker
- Ukas shoutout
- Selvstendig jobbing resten av tiden