Auto complete

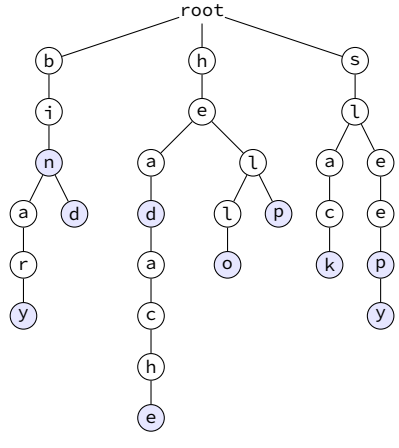
De fleste teksteditorer har funksjonalitet for «auto completion», altså der det dukker opp en liten boks med mulige måter å fullføre et påbegynt ord. Nedenfor gir vi to mulige datastrukturer som kan brukes for å implementere en enkel mekanisme for å fullføre et påbegynt ord.

Vi viser eksempler på hvordan datastrukturene ser ut dersom de lagrer ordene: `bin`, `binary`, `bind`, `head`, `headache`, `hello`, `help`, `slack`, `sleep`, `sleepy`.

Strategi 1

Den første strategien er å bruke et *prefiks-tre* som holder på strenger. Et prefiks-tre har en rot. Hvis en streng s er lagt inn i treet, har rotnoden et barn med s sin første bokstav. Den noden har igjen en peker til s sin andre bokstav, og så videre. Den siste bokstaven i s er markert som en *terminal* node (terminalnodene er fargelagt blå i eksempelet). Det vil si at hvis du følger en sti fra rotnoden til en terminalnode, så har du stavet et ord som er lagt inn i treet.

Fra et prefiks-tre er det enkelt å finne alle måter å fullføre et påbegynt ord. Det gjøres ved å returnere alle stier fra rotnoden som begynner med bokstavene fra et gitt ord og ender i en terminalnode.



Strategi 2

Den andre strategien er å bruke et hashmap, der vi lar hver prefiks av et ord mappe til en liste av alle mulige måter å fullføre ordet. Hvis `H` er et hashmap, og s er en streng som skal legges til, ser vi på hver prefiks p av s og legger s til i listen `H[`p`]`. I eksempelet ved siden av ser dere hvordan hashmapet kan se ut dersom alle eksempelordene er lagt (merk at noen av listene er kortet ned, indikert ved «...»).

Fra et slikt hashmap er det svært enkelt å finne alle måter å fullføre et påbegynt ord. Det gjøres ved å slå opp det påbegynte ordet i hashmapet og returnere listen.

Nøkkel	Verdi
`""`	`"bin", ..., "sleepy"`
`"b"`	`"bin", "binary", "bind"`
`"bi"`	`"bin", "binary", "bind"`
`"bin"`	`"bin", "binary", "bind"`
`"bina"`	`"binary"`
`"binar"`	`"binary"`
`"binary"`	`"binary"`
`"bind"`	`"bind"`
`"h"`	`"head", ... "help"`
`"he"`	`"head", ... "help"`
`"hea"`	`"head", "headache"`
`"head"`	`"head", "headache"`
`"headа"`	`"headache"`
`"headac"`	`"headache"`
`"headach"`	`"headache"`
`"headache"`	`"headache"`
`"hel"`	`"hello", "help"`
`"hell"`	`"hello"`
`"hello"`	`"hello"`
`"help"`	`"help"`
`"s"`	`"slack", "sleep", "sleepy"`
`"sl"`	`"slack", "sleep", "sleepy"`
`"sla"`	`"slack"`
`"slac"`	`"slack"`
`"slack"`	`"slack"`
`"sle"`	`"sleep", "sleepy"`
`"slee"`	`"sleep", "sleepy"`
`"sleep"`	`"sleep", "sleepy"`
`"sleepy"`	`"sleepy"`

Drøft fordeler og ulemper ved de to ulike strategiene både med tanke på kjøretid og minnebruk