

2D1320, Tilda, Tentamenslösning 16 december 2000

1. *Palindromsyntax*

$$\langle \text{palindrom} \rangle ::= \varepsilon \mid A \mid B \mid A \langle \text{palindrom} \rangle A \mid B \langle \text{palindrom} \rangle B$$
2. *Palindromsortering*

Bloomfilter är oanvändbart eftersom det bara kan användas för att undersöka om ett ord finns med eller ej.

Distributionsräkning är effektivast ($O(N)$ jämförelser) eftersom orden ska sorteras efter längd, och det bara finns 22 olika längder (PORTUGALRALLARLAGUTROP är ju den längsta palindromen).

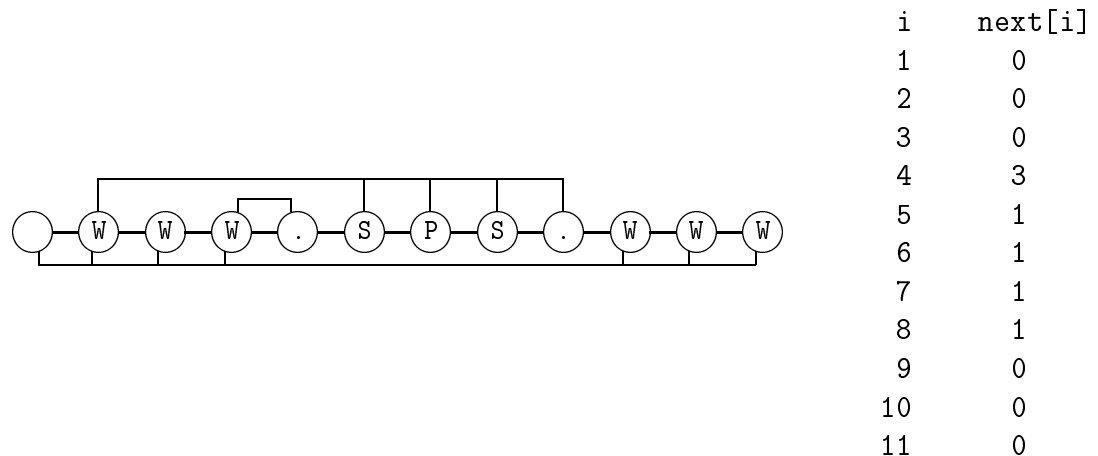
Mergesort och quicksort kommer på andra plats med $O(N \log N)$.

Insättningsortering tar lite längre tid med $O(N^2)$.

3. *Palindromhashning*

Om det tjugofyrasiffriga talet är en `String` `tal` kan man först vända det bakfram och beräkna `tal.hashCode()+tal.hashCode() % p`, där `p` är ett primtal cirka 1.5 gånger antalet existerande streckkoder.

För passerkorten kan man använda bloomfilter, eftersom man bara behöver veta om koden finns med i databasen eller ej.

4. *Palindromisk webbadress*5. *Rekursiv palindrom*

Vi använder djupetförstökning. Metoden `makeWord` får skapa nya ord genom att lägga till samma bokstav sist och först i ordet. Man kan använda Viggos bloomfilter för att undersöka om ordet finns eller inte. Ordposterna måste ha en `writeChain` så att man kan skriva ut den rekursiva palindromen med en rekursiv `writeChain`-metod. Då vi upptäcker att det inte går att bygga på ordet längre sparar vi en referens till det om ordet är längre än det tidigare sparade längsta ordet. Dumsöner behöver vi inte bekymra oss om.

6. *Palindromträd*

Låt `speglad(root1,root2)` vara `true` om båda träden är tomma, `false` om bara det ena är tomt och för övrigt samma som `speglad(root1.left,root2.right)&&speglad(root1.right,root2.left)`

7. *Palindromstackar*

1. Läs in varje bokstav och pusha på stacken, räkna samtidigt antalet bokstäver.
2. Poppa hälften av bokstäverna och pusha på en annan stack (om antalet bokstäver var udda - släng då den mittersta).
3. Poppa en bokstav från vardera stacken och jämför om dom är lika.
4. Avbryt när bokstäverna inte längre är lika (då var det ingen palindrom) eller när stackarna tar slut (då var det en palindrom!)

8. *Verbal magi i gamla brev*

Olika forskare kan tänkas behöva olika information om breven. Vissa kanske behöver titta på brevens utseende (då är en `Image` bra), andra behöver bara brevens text (då passar en `String`), och åter andra behöver veta hur texten brutits på rader (då är en `String[]` lämplig).

Om man använder en abstrakt datatyp behöver man inte skriva om forskarnas program varje gång en forskare får behov av någon information som inte enkelt kan fås fram ur den aktuella brevrepresentationen.

Några metoder för den abstrakta brevtypen:

```
Date datum();
String adressat();
String text();
String rad(int nummer);
int antalRader();
Image bild();
```