HTML poeng

Du har tatt på deg en jobb hos en bedrift som har en eldre nettside skrevet i ren HTML. Nylig har de fått nyss om at nettsiden fungerer svært dårlig, og de (som har lite intern kompetanse på slikt) har gitt opp og håper at den unge lovende informatikeren skal finne ut av hva i all verden som foregår. Nettsiden består av flere tusen håndskrevne HTML-sider; all snakk om å «lage noe nytt og fresht» blir møtt med spørrende blikk.

Det tar ikke spesielt lang tid før du finner en feil som gjentar seg i mange av sidene det er rapportert feil på. Mange av de håndskrevne HTML-filene er feilformaterte! Du merker deg at det mange steder åpnes en `<div>`-tag, som aldri lukkes med en `</div>`. Andre steder lukkes det en `</div>`-tag, der den aldri har blitt åpnet. Her er et eksempel på en feilformatert HTML-fil:

```
<div>
  <div>
    Velkommen til nettsiden vår!
  <div>
   Se våre nyeste saker!
    <div>
    Vi har ansatt en ung person! Kanskje blir nettsiden oppdatert nå?
    FØLG MED PÅ UTVIKLINGEN. Hilsen Kåre.
    </div>
</div>
```

(a) 4 poeng

Du innser at det første du må gjøre er å finne alle filene som inneholder feil. Etter litt Unix-magi sitter du igjen med filer der all tekst bortsett fra `<div>` og `</div>` er fjernet. Alt går på skinner frem til du må løse en liten nøtt:

> *Hvordan sjekker man at det for hver tag som åpnes med `<div>` også lukkes med `</div>`, og at ingen tag lukkes før den er åpnet?*

Du må skrive en prosedyre `GoodDivs` som sjekker dette, med andre ord, at `<div>`-tagene er velformaterte. Den tar et array `A` som argument, der hvert element enten er strengen `"<div>"` eller strengen `"</div>"`.

ALGORITHM: AVGJØRE OM `<div>`-TAGENE ER VELFORMATERTE

Input: Et array med `"<div>"` og `"</div>"` av lengde n
Output: Returner **true** hvis tag-ene er velformaterte, **false** ellers

```
1 Procedure GoodDivs(A)
    // Fyll ut
```

Dette er en forholdsvis enkel oppgave, som først og fremst var ment som en oppvarming til oppgave (b). Merk at et (riktig) svar på (b) også egentlig er et riktig svar på denne oppgaven.

Den enkleste måten å gå frem på her er å løpe gjennom alle tagene og øke en teller ved `"<div>"` og senkes ved `"</div>"`. Dersom telleren på noe tidspunkt blir negativ kan vi returnere **false**. Etter alle tag-ene er løpt gjennom må vi sjekke at telleren er 0.

```
1  Procedure GoodDivs(A)
2      i ← 0
3      for tag in A do
4          if tag = "<div>" then
5              i ← i + 1
6          else
7              i ← i − 1
8          if i < 0 then
9              return false
10     return i = 0
```

Veiledende poengfordeling:

- Totalt 1 poeng for løsninger som sjekker direkte at antall == antall
- Totalt 1 poeng for løsninger som sjekker at alle etterfølges av en direkte
- Minus 2 poeng for å glemme å sjekke om antall noen gang overstiger antall
- Minus 1 poeng for å glemme å sjekke om vi ender opp med likt til slutt

(b) 8 poeng

`GoodDivs` fungerer som den skal og finner mange filer som inneholder feilformatert HTML. Dessverre oppdager den ikke alle filene det er rapportert feil på. Du observerer at de inneholder samme feil, men med andre type tag-er! Altså er det ikke bare `<div>`-tagene som fører til problemer, men også andre tag-er som `<head>`, `<body>`, `<p>` og flere andre. Du finner ut at prosedyren `GoodDivs` må generaliseres til en prosedyre `GoodTags`.

> *Hvordan sjekker man at det for hver tag som åpnes også lukkes med en tag av samme type, og at ingen tag lukkes før den er åpnet?*

Du har allerede en enkel prosedyre `isOpen`, som tar en streng og returnerer **`true`** hvis det er en åpne-tag og **`false`** hvis det er en lukke-tag. Altså vil `isOpen("<tag>")` returnere **`true`** og `isOpen("</tag>")` returnere **`false`**. I tillegg har du en prosedyre som gir deg hvilken type en tag har; for eksempel vil `TagType("<div>")` gi `"div"`, og `TagType("</p>")` gir `"p"`.

ALGORITHM: AVGJØRE OM TAG-ENE ER VELFORMATERTE

Input: Et array `A` med tag-er av lengde n
Output: Returner **`true`** hvis tag-ene er velformaterte, **`false`** ellers

```
1 Procedure GoodTags(A)
  |  // Fyll ut
```

Hint: Her kan et lurt valg av datastruktur være til stor hjelp.

Oppgaveteksten åpner for at `["<div>", "<p>", "</div>", "</p>"]` teller som velformatert. Dette var ikke intensjonen. Løsninger som teller dette som velformatert vil også kunne gi full uttelling. Det kan enkelt gjøres ved å ha et map fra tag-type til en teller, som brukes på samme måte som forrige deloppgave.

Løsningen som følger intensjonen (som faktisk vil validere balanserte HTML-filer) kan se slik ut:

```
1 Procedure GoodDivs(A)
2   s ← empty stack
3   for tag ∈ A do
4     if isOpen(tag) then
5       s.push(tag)
6       continue
7     if s is empty or TagType(s.pop()) ≠ TagType(tag) then
8       return false
9   return s is empty
```

Her vektes poeng og trekk vanligvis på samme måte som i (a).