TÖL403G GREINING REIKNIRITA

15. Strengleit 1

Hjálmtýr Hafsteinsson
Vor 2022

Í þessum fyrirlestri

- Venjuleg leit (*brute force*)
  - Tímaflækja
- Strengir sem tölur
  - Strengleit sem samanburður á tölum
- Reiknirit Karp-Rabin
  - Slembin útgáfa
- Óþarfir samanburðir (*redundant comparisons*) í strengleit
  - Getum sparað okkur tiltekna samanburði

DC 7.1 − 7.4

Verkefni

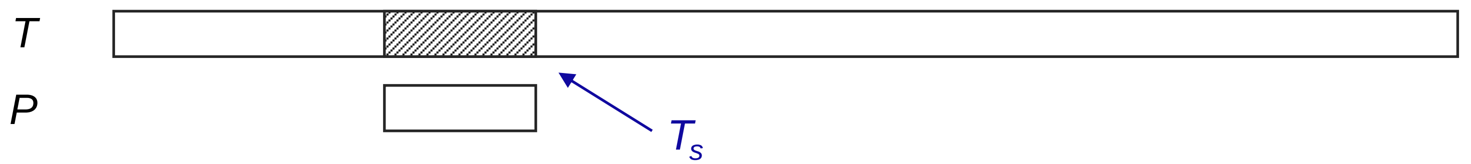
- Vinnum með <u>strengi</u> (*strings*), sem er runa af <u>bókstöfum</u>
  - Bókstafirnir koma úr <u>stafrófi</u> (*alphabet*) Σ
  - Σ gæti verið ASCII tákn, Unicode tákn, {A, C, G, T}, ...
- Verkefnið:

Notkunarsvið:
- Textavinnsla
- Merkjafræði
- Gagnaþjöppun
- Líffræði
- Efnafræði

Gefnir tveir strengir: <u>texti</u> $T[1..n]$ og <u>mynstur</u> (*pattern*) $P[1..m]$.

Finna fyrsta hlutstreng í textanum sem er eins og mynstrið

Gerum ráð fyrir því að m << n



Skilum staðsetningu P í T, eða því að mynstrið sé ekki í T (gætum t.d. skilað 0 þá)

Látum T_s tákna hlutstrenginn $T[s..s+m-1]$

Þá viljum við finna minnstu hliðrun (*shift*) s þannig að $T_s = P$

Venjulegt (*brute force*) reiknirit

- Augljóst reiknirit ber P saman við alla mögulega hlutstrengi T_s, $s=1,\ldots n-m+1$

```
ALMOSTBRUTEFORCE(T[1..n], P[1..m]):
  for s ← 1 to n − m + 1
      equal ← TRUE
      i ← 1
      while equal and i ≤ m
          if T[s + i − 1] ≠ P[i]
              equal ← FALSE
          else
              i ← i + 1
      if equal
          return s
  return NONE
```

Fyrir alla mögulega *m*-stafa hlutstrengi í *T*

Bera saman við *P*, staf fyrir staf

Hætta í innri lykkju um leið og stafir passa ekki

Reyndar er þetta reiknirit ekki svo slæmt á "slembnu inntaki", en hvenær er það?!

Versta tilfelli:

T: "AAA...AAA"
P: "A...AB"

Hér klárast allir hlutstrengssamanburðir og fjöldi þeirra er $O(nm)$

[Sýnidæmi um virkni]()

Strengir sem tölur

- Skoðum aðra nálgun á verkefnið: hlutstrengir sem tölur

Gerum ráð fyrir að stafrófið Σ sé {0, 1, ..., 9}

Getum þá litið á *P* sem heiltöluna $$p = \sum_{i=1}^{m} 10^{m-i} \cdot P[i]$$

og líka hlutstrenginn T_s sem $$t_s = \sum_{i=1}^{m} 10^{m-i} \cdot T[s+i-1]$$

<u>Dæmi:</u>

T: "3141592653589"

og *P*: "9265"

Þá er p = 9265

og t_1 = 3141, t_2 = 1415, ...

Nú er verkefnið að finna fyrsta *s* þannig að heiltalan *p* = heiltalan t_s

Útreikningur á heiltölum

Við getum reiknað p í $O(m)$ reikniaðgerðum með reglu Horners:

$$p = P[m] + 10\big(P[m-1] + 10\big(P[m-2] + \cdots + 10\big(P[2] + 10 \cdot P[1]\big)\cdots\big)\big)$$

<u>Dæmi:</u> P = "9265" $p = 5 + 10\cdot(6 + 10\cdot(2 + 10\cdot 9)) = 9265$

T: "3141592653589"
P: "9265"

Við getum reiknað t_s á svipaðan hátt og p, en við getum nýtt okkur gildi t_s til að finna t_{s+1}

$$t_{s+1} = 10\big(t_s - 10^{m-1} \cdot T[s]\big) + T[s+m]$$

<u>Dæmi:</u>

Reiknum $t_1 = 3141$, viljum nú finna t_2

þá er $t_2 = 10\cdot(3141 - 3\cdot 1000) + 5$ $= 10\cdot 141 + 5$ $= 1410 + 5$ $=$ <u>1415</u>

Taka burt efsta tölustaf

Hliðra afgangi upp

Bæta nýjum tölustaf aftast

Talnaleit

$\underline{\textsc{NumberSearch}(T[1..n], P[1..m]):}$

$\sigma \leftarrow 10^{m-1}$

$p \leftarrow 0$
$t_1 \leftarrow 0$
for $i \leftarrow 1$ to m
    $p \leftarrow 10 \cdot p + P[i]$
    $t_1 \leftarrow 10 \cdot t_1 + T[i]$
for $s \leftarrow 1$ to $n - m + 1$
    if $p = t_s$
        return s
    $t_{s+1} \leftarrow 10 \cdot \left(t_s - \sigma \cdot T[s]\right) + T[s+m]$
return None

Forreiknum 10^{m-1}, því það er notað á mörgum stöðum

Reikna fyrst p and t_1 frá grunni á $O(m)$

Bera saman p og t_s

Reikna t_{s+1} út frá t_s

Virðumst geta leyst verkefnið á $O(n+m)$ tíma

Þetta er þó ekki alveg rétt. Tölurnar p og t_s eru m tölustafir og ekki raunhæft að aðgerðir á þær taki $O(1)$ tíma.

Raunhæfari tímaflækja er $O(nm)$

Æfing

- Í *Talnaleit* notum við eftirfarandi formúlu til að finna t_{s+1} út frá t_s á $O(1)$ tíma:

$$t_{s+1} = 10\left(t_s - 10^{m-1} \cdot T[s]\right) + T[s+m]$$

- Hver væri formúlan ef við ætluðum að finna t_{s-1} út frá t_s á $O(1)$ tíma?

Til dæmis: Fara frá $t_2 = 1415$ í $t_1 = 3141$

T: "3141592653589"

P: "9265"

Reiknirit Karp-Rabin

Notar reikniritið ***NumberSearch***, en ...

framkvæmir alla útreikninga módúlus q

q < 400 milljón, 8-9 stafa frumtala

q er frumtala og valin þannig að $10 \cdot q$ komist fyrir í venjulegri heiltölubreytu

Köllum gildin (p **mod** q) og (t_s **mod** q) fingraför (*fingerprints*)

Getum áfram reiknað (p **mod** q) og (t_s **mod** q) á $O(m)$ tíma

og (t_{s+1} **mod** q) út frá (t_s **mod** q) á $O(1)$ tíma

Þá höfum við:

Ef (p **mod** q) ≠ (t_s **mod** q) þá er $P \neq T_s$

EN ...

Ef (p **mod** q) = (t_s **mod** q) þá vitum við ekki hvort $P = T_s$ eða ekki!

Notum þá bara venjulega reikniritið til að bera saman P og T_s

Sýnidæmi um Karp-Rabin

T: "3141592653589"

P: "9265"

Veljum *q* sem 13

Þetta er þá fingrafarið sem við erum að leita að

Þar sem *p* = 9265 þá er $\tilde{p} = (p \;\mathbf{mod}\; 13)$ = (9265 **mod** 13) = **9**

Við höfum svo $\widetilde{t_1} = (t_1 \;\mathbf{mod}\; 13)$ = (3141 **mod** 13) = **8**

$\widetilde{t_2} = (t_2 \;\mathbf{mod}\; 13)$ = (1415 **mod** 13) = **11**

$\vdots$ $\vdots$

$\widetilde{t_6} = (t_6 \;\mathbf{mod}\; 13)$ = (9265 **mod** 13) = **9** **Passar!** og $P = T_6$

$\widetilde{t_8} = (t_8 \;\mathbf{mod}\; 13)$ = (6535 **mod** 13) = **9** **Passar!** en hér er $P \neq T_8$

Í hvert sinn sem fingraförin passa þá þarf að bera saman strengina

Tímaflækja

- Tíminn á reikniritinu er $O(n + Fm)$
  - þar sem F er fjöldi falskra parana (fingraför passa en strengir ekki)

- Fingraförin t_s ættu að hoppa um á bilinu 0 til $q-1$ nokkuð slembið
- Í fljótu bragði sýnist því að líkurnar á fölskum pörunum ættu að vera $1/q$
- Það þýðir að $F = n/q$ "að meðaltali" og keyrslutíminn því $O(n + nm/q)$

Ef við veljum $q \geq m$, þá er þessi tími $O(n)$

En athugið að það er ekkert slembið í reikniritinu.
Þessi "meðaltími" byggir algerlega á dreifingu inntaksins!

Slembið Karp-Rabin reiknirit

- Velja frumtöluna q á slembinn hátt

```
KARPRABIN(T[1..n], P[1..m]):
    q ← a random prime number between 2 and ⌈m² lg m⌉
    σ ← 10^(m−1) mod q
    p̃ ← 0
    t̃_1 ← 0
    for i ← 1 to m
        p̃ ← (10 · p̃ mod q) + P[i] mod q
        t̃_1 ← (10 · t̃_1 mod q) + T[i] mod q
    for s ← 1 to n − m + 1
        if p̃ = t̃_s
            if P = T_s        ⟨⟨brute-force O(m)-time comparison⟩⟩
                return s
        t̃_{s+1} ← (10 · (t̃_s − (σ · T[s] mod q) mod q) mod q) + T[s + m] mod q
    return NONE
```

Útreikningur á næsta $\tilde{t}_s$

Frumtölusetningin:

Fjöldi frumtalna < u er $\Theta(u / \log(u))$

Með því að velja q úr mengi $m^2 \cdot \log(m)$ talna þá eru þar $\Theta(m^2)$ frumtölur

Þetta gefur okkur að líkurnar á falskri pörun í t_s er $O(1/m)$

Væntur fjöldi falskra parana er O(n/m)

Væntur tími á Karp-Rabin er $O(n)$

Önnur slembin útgáfa

- Kemur í ljós að það er ekki auðvelt að finna slembi frumtölu
  - Getum búið til slembi heiltölu og athugað hvort hún sé frumtala
  - Ættum að finna eina slíka eftir ~ log(m) ítranir
  - Það kostar nokkuð að athuga hvort stór heiltala sé frumtala

- Einfaldari leið:

Við höfum notað grunntöluna 10, en við gætum haft annað gildi b

Velja grunntölu talnakerfisins af handahófi

Ekki slembin, gætum notað fast gildi á q

Veljum fyrst frumtölu $q > m^2$

Veljum svo grunntöluna b af handahófi á bilinu 2 til $q-1$

Reiknum svo: $$p(b) = \sum_{i=1}^{m} b^i \cdot P[m-i]$$

svipað fyrir $t_s(b)$

Fáum hér líka að væntur fjöldi falskra parana er O(n/m)

Væntur tími er því $O(n)$

Óþarfir samanburðir (*redundant comparisons*)

- Förum aftur í að bera saman einstaka stafi í strengjunum

Segjum að við höfum mynstrið *P* = "ABRACADABRA"

Berum það saman við langan textastreng:

T: ....*XXX*ABRABRA....

P: ABRA~~C~~ADABRA

Búin að bera saman A, B, R, A og þau passa öll ...

... en svo kemur B í textanum og C í mynstrinu

Venjulega reikniritið myndi nú færa mynstrið um eitt sæti (þ.e. hækka *s* um 1) og byrja aftur á byrjun mynstursins

T: ....*XXX*ABRABRA....

P: ABRACADABRA

En við höfum þegar séð þennan staf í textanum (B) og vitum að hann passar ekki við *P*[1], sem er A

Óþarfir samanburðir, frh.

T: ....*XXX*ABRABRA....
P: ABRA~~C~~ADABRA

Við höfum parað saman 4 stafi og viljum nota okkur þá þekkingu í framhaldinu

Vitum að *P*[1] (A) passar ekki við næsta staf (B)

Heldur ekki við þar næsta staf (R)

En *P*[1] (A) passar við síðasta stafinn (A)

En við vitum þetta!!

Þurfum því ekki að bera þá saman!

Næsti samanburður er á *P*[2] og næsta staf textans (B)

T: ....*XXX*ABRABRA....
P: ABRACADABRA

Sýnidæmi

T: ....ABRABRDABRACA....

P: ABR ABR A ABRACADABRA

B ≠ C Passar ekki - hliðra mynstri

Halda áfram þar sem passar

D ≠ A Passar ekki - hliðra mynstri

Verðum hér að byrja frá byrjun

D ≠ A Passar ekki - hliðra mynstri

Verðum hér líka að byrja frá byrjun

Athugið:

Textabendirinn ↓ færðist alltaf áfram eða stóð í stað. Hann fór aldrei til baka

Losna við samanburði

- Ef við hugsum samanburðina út frá sjónarhóli textans:

Við þurfum ekki að bera stafina ABRA aftur við stafi í mynstrinu

Við erum búin að bera þá saman einu sinni og getum nýtt okkur þá þekkingu

Fyrir hvern staf *T*[i] í textanum:

Ef hann passar við einhvern staf í mynstrinu þá
þarf aldrei að bera hann saman við neinn annan staf

Ef hann passar ekki, þá gæti þurft að bera hann saman aftur

Þurfum aldrei að
bakka í textanum

Eina spurningin er:

Hversu langt á að hliðra mynstrinu eftir stafir passa ekki?

Knuth-Morris-Pratt
reikniritið segir okkur það!

Fyrirlestraæfingar

1. Hversu marga samanburði þarf venjulega reikniritið (*brute force*) til að finna mynstrið "0001" í textanum "0010000001"? Hér er m=4 og n=10
2. Höfum að T = "314159..." og það er búið að reikna t_2 (= 1415). Finnið gildið á t_3 út frá t_2 og T[6].
3. T = "27182818", P = "28", q er 5. Reiknið (p **mod** 5) og (t_s **mod** 5) fyrir s = 1, ..., 7. Hvar passa fingraförin og hvar er röng pörun (*false match*)?