Selection

Tölvunarfræði 2, vor 2012

Hallgrímur H. Gunnarsson

Háskóli Íslands

2012-03-02

Selection

Markmið:

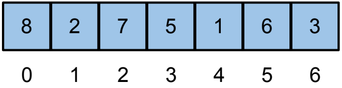
$\boxed{\text{Finna k-ta minnsta (eða k-ta stærsta) gildið í óröðuðu fylki af stærð N}}$

Dæmi:

- minnsta gildið (k=0)
- stærsta gildið (k=N-1)
- miðgildið (k=N/2)
- fjórðungsmörk (25%, 50%, 75%)

Selection

Hvað gildir um k-ta minnsta gildið í fylki? (teljum frá $k=0$)

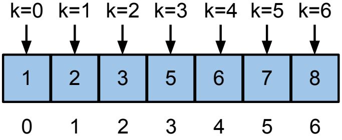


k=0 er .. ?

k=1 er .. ?

k=2 er .. ?

k-ta minnsta gildið er í sæti k þegar fylkið er í vaxandi röð



Select fall

Skilgreinum eftirfarandi fall sem leysir selection.

Fallið gerir aðeins meira en að finna k-ta minnsta, það skiptir fylkinu líka þ.a. öll minni gildi fari fyrir neðan og öll stærri fyrir ofan

```
// Notkun: select(f,i,j,k)
// Fyrir:  f[i..j-1] er svæði í fylkinu f, i <= k < j
//         f[i..j-1] má vera óraðað.
// Eftir:  Búið er að víxla gildunum í f[i..j-1] þannig
//         að f[i..k-1] <= f[k] <= f[k+1..j-1].
static void select(int[] f, int i, int j, int k)
```

Select fall

Eftirskilyrði fallsins segir að f[k] er í réttu sæti m.v. vaxandi röð:

```
f[i..k-1] <= f[k] <= f[k+1..j-1]
```

Um fylki í vaxandi röð gildir:

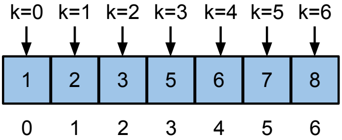
```
f[i] <= f[i+1] <= .. <= f[k] <= f[k+1] <= .. <= f[j-1]
```

Þá eru öll gildi í sínum endanlegu sætum. Seinni fullyrðingin er sterkari og felur í sér fyrri fullyrðinguna sem sértilfelli.

Ein leið til að uppfylla eftirskilyrðið er því að uppfylla sterkara skilyrðið, þ.e. raða fylkinu, en það krefst augljóslega meiri vinnu enda er gengið lengra en þörf krefur.

Lausn 1 – Röðum fylkinu

k-ta minnsta gildið er í sæti k þegar fylkið er í vaxandi röð



Einfaldasta lausnin fyrir selection: röðum fylkinu í vaxandi röð

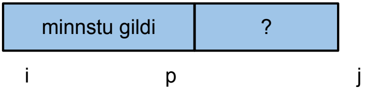
Kostar O(N log N) en erum þá búin að leysa selection fyrir öll k

Eftir röðun: k-ta minnsta gildið er f[k]

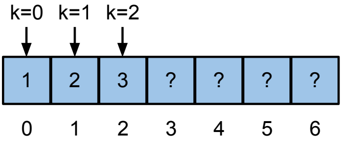
Hvað ef við höfum bara áhuga á einu k, en ekki öllum?

Lausn 2 – Röðum fylkinu upp að k

Upprifjun: fastayrðing fyrir ytri lykkju selection sort:



Önnur lausn: röðum fylkinu upp að k+1 með selection sort



Kostar O(k*N) en erum þá búin að leysa fyrir öll fyrstu k

Lausn 2 – Röðum fylkinu upp að k

```
// Notkun: select(f,i,j,k)
// Fyrir:  f[i..j-1] er svæði í fylkinu f, i <= k < j
//         f[i..j-1] má vera óraðað.
// Eftir:  Búið er að víxla gildunum í f[i..j-1] þannig
//         að f[i..k-1] <= f[k] <= f[k+1..j-1].
static void select(int[] f, int i, int j, int k)
{
    ssort(f, 0, k+1);
    // f[i..k] inniheldur minnstu gildi úr f[i..j-1]
    // í vaxandi röð
}
```

Betri lausn

Það er auðvelt að leysa minnsta (k=0) og stærsta (k=N-1) á línulegum tíma.

En hvað með eitthvað annað k?

Rifjum upp quicksort:

- Quicksort notar skiptingu (partition)
- Skiptir fylkinu eftir e-u gildi (vendistak/pivot)
- Það tiltekna gildi endar í sínu lokasæti eftir skiptingu
- Öll minni gildi fyrir neðan, öll stærri fyrir ofan
- **Við vitum ekki hvar vendistakið endar fyrr en eftir skiptingu**

Skipting í quicksort

Við vitum ekki hvar vendistakið endar fyrr en eftir skiptingu

Stóri vandinn í quicksort er að við vitum ekki fyrirfram besta vendistakið, þ.e. gildið sem mun gefa okkur jafna skiptingu.

Það verður því að velja vendistakið eftir öðrum leiðum.

Til ýmis afbrigði af quicksort eftir því hvernig það er gert, t.d. slembið val, alltaf fyrsta, miðgildi þriggja, o.s.frv.

Select og skipting

Í skiptingunni í quicksort er skipt eftir tilteknu gildi. Við vitum ekki hvar gildið endar fyrr en eftir skiptinguna.

Í select erum við að leita eftir gildi sem endar í tilteknu sæti k eftir skiptingu, sem er þá k-ta minnsta gildið.

Getum við notað partition fallið til að útfæra select?

Útfærum select með skiptingu

Skiptum f[i..j-1] eftir fremsta gildinu:

- Gildið endar í sæti n eftir skiptingu
- Ef $k == n$ þá erum við búin að leysa verkefnið
- Ef $k < n$, leysum þá sama verkefni endurkvæmt fyrir f[i..n]
- Ef $k > n$, leysum þá sama verkefni endurkvæmt fyrir f[n+1..j]

Dæmi

Höfum fylki f[0..10], viljum finna k=3:

62	88	59	94	4	71	26	17	44	28	37
0	1	2	3	4	5	6	7	8	9	10

Fylkið í röð:

4	17	26	28	37	44	59	62	71	88	94
0	1	2	3	4	5	6	7	8	9	10

Við viljum enda með skiptingu þar sem:

- f[0..2] <= 28 f[3] == 28 f[4..10] >= 28

Dæmi – 1. umferð

Fyrir skiptingu:

62	88	59	94	4	71	26	17	44	28	37
0	1	2	3	4	5	6	7	8	9	10

Eftir skiptingu:

37	59	4	26	17	44	28	62	71	94	88
0	1	2	3	4	5	6	7	8	9	10

Gildið endaði í sæti 7. Staðan eftir skiptingu:

- f[0..6] $<=$ 62 f[7] $==$ 62 f[8..10] $>=$ 62

Dæmi – 2. umferð

Fyrir skiptingu:

0	1	2	3	4	5	6	7	8	9	10
37	59	4	26	17	44	28	62	71	94	88

Eftir skiptingu:

0	1	2	3	4	5	6	7	8	9	10
28	4	26	17	37	44	59	62	71	94	88

Gildið endaði í sæti 4. Staðan eftir skiptingu:

- f[0..3] <= 37 f[4] == 37 f[5..10] >= 37

Dæmi – 3. umferð

Fyrir skiptingu:

28	4	26	17	37	44	59	62	71	94	88
0	1	2	3	4	5	6	7	8	9	10

Eftir skiptingu:

17	4	26	28	37	44	59	62	71	94	88
0	1	2	3	4	5	6	7	8	9	10

Gildið endaði í sæti 3. Verkefnið leyst! Staðan eftir skiptingu:

- f[0..2] $<=$ 28   f[3] $==$ 28   f[4..10] $>=$ 28

Tímaflækja quickselect

Quickselect tekur O(N) tíma að jafnaði.

Rökstuðningur: hver skipting skiptir fylkinu í uþb. jafna helminga

Fáum þá: $N + N/2 + N/4 + .. + 1 < 2N$ sem er O(N)