Helmingunarleit

Tölvunarfræði 2, vor 2012

Hallgrímur H. Gunnarsson

Háskóli Íslands

2012-02-15

Tímaflækja

Hvernig er hægt að tala almennt um að eitt reiknirit sé hraðvirkara en annað?

Við viljum nothæfa skilgreiningu á hraða sem er:

- Óháð forritunarmáli, þýðanda og tegund tölvu sem keyrt er á
- Óháð sértilvikum gagna

Skilgreining á Big O

$O(g(x))$ eru efri mörk (e. upper bound, worst case) fyrir $f(x)$

$$f(x) \in O(g(x)) \quad \text{eða} \quad f(x) = O(g(x)) \tag{1}$$

ef það er til fasti M og tala x_0 þ.a.

$$|f(x)| \leqslant M|g(x)| \tag{2}$$

fyrir öll $x > x_0$

Big Oh – Dæmi

Grf.

$$f(x) = 6x^4 - 2x^3 + 5 \tag{3}$$

Notum skilgreininguna á Big O og fáum:

$$\begin{aligned} |6x^4 - 2x^3 + 5| &\leqslant 6x^4 + |2x^3| + 5 \qquad (4) \\ &\leqslant 6x^4 + 2x^4 + 5x^4 \qquad (5) \\ &\leqslant 13x^4 \qquad (6) \\ &\leqslant 13|x^4| \qquad (7) \end{aligned}$$

þ.a.

$$f(x) = O(x^4) \tag{8}$$

Línuleg leit í fylki

```
// Notkun: k = search(f,i,j,x);
// Fyrir:  f[i..j-1] er svæði í f
// Eftir:  i <= k <= j,
//         ef x er til í f[i..j-1] þá er f[k] == x
//         en annars k == -1
int search(double f[], int i, int j, double x) {
  for (int k = i; k < j; k++ ) {
    // x er ekki í f[i..k-1]
    if (f[k] == x)
      return k;
  }
  return -1;
}
```

Stærð verkefnisins er $j - i = N$. Ef $T(N)$ er tíminn sem það tekur að leysa vandamálið þá er ljóst að $T(N)$ er af stærðargráðunni $O(N)$

Getum við gert betur?

Hvað ef við megum gera ráð fyrir að fylkið sé í vaxandi röð?

Hvernig leitar maður í símaskrá eða atriðaorðaskrá?

Línuleg leit í fylki – V2

```
// Notkun: k = search(f,i,j,x);
// Fyrir:  f[i..j-1] er í vaxandi röð
// Eftir:  f[i..j-1] er óbreytt, i <= k <= j, og
//         f[i..k-1] <= x < f[k..j-1]
//
// Ath. f[k] er fyrsta talan sem er > x,
//      f[k-1] er síðasta talan sem er <= x.
int search(double[] f, int i, int j, double x) {
  for (int k = i; k < j; k++) {
    // | <= x | óþekkt |
    //  i      k        j
    if (f[k] > x)
      return k;
  }
  return j;
}
```

Aðeins almennari aðferð. Finnur réttan stað í fylkinu.

Helmingunarleit

Demo

Helmingunarleit

```
// Notkun: k = search(f,i,j,x);
// Fyrir:  f[i..j-1] er í vaxandi röð
// Eftir:  f[i..j-1] er óbreytt, i <= k <= j, og
//         f[i..k-1] <= x < f[k..j-1]
//
// Ath. f[k] er fyrsta talan sem er > x,
//      f[k-1] er síðasta talan sem er <= x.
int search(double[] f, int i, int j, double x) {
  if (i == j) return i;
  int m = (i+j)/2;
  if (f[m] > x)
    return search(f,i,m,x);
  else
    return search(f,m+1,j,x);
  }
```

Greining á helmingunarleit

Hversu oft er hægt að helminga N ?

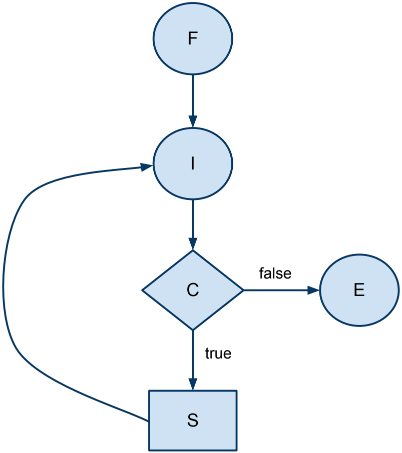
Tíminn sem tekur að leita með helmingunarleit í fylki af stærð N er af stærðargráðunni $O(\log N)$

N	$\log N$
1000	10
10.000	13
100.000	16
1.000.000	19
10.000.000	23
100.000.000	26

Fastayrðing lykkju

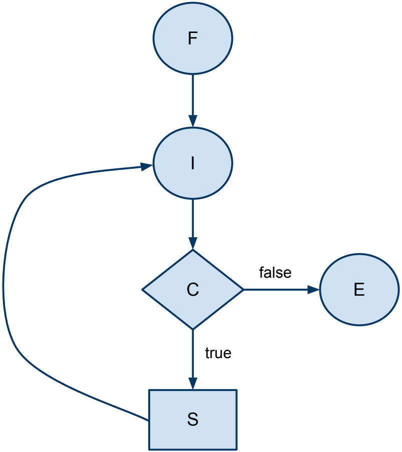
```
// F
while (C)
{
    // I
    S;
}
// E
```

- F er forskilyrði lykkjunnar
- C er lykkjuskilyrði
- I er fastayrðing lykkjunnar
- S er forritstexti
- E er eftirskilyrði lykkjunnar



Fastayrðing lykkju

```
// F
while (C)
{
    // I
    S;
}
// E
```



Reglur:

1. F → I
2. {I og C} S {I}
3. I og ¬C → E

Helmingunarleit

```
// Notkun: k = search(f,i,j,x);
// Fyrir:  f[i..j-1] er í vaxandi röð
// Eftir:  f[i..j-1] er óbreytt, i <= k <= j, og
//         f[i..k-1] <= x < f[k..j-1]
int search(double[] f, int i, int j, double x) {
  int p=i, q=j;
  while (p != q) {
    // | <= x | óþekkt | > x |
    //  i      p        q     j
    int m = (p+q)/2;
    if (f[m] <= x)
      p = m+1;
    else
      q = m;
  }
  // | <= x | > x |
  //  i      p     j
  return p;
}
```

Er forritið rétt?

Helmingunarleit er einföld en það getur verið snúið að ná öllu rétt:

- Jon Bentley setti helmingunarleit sem verkefni í námskeiði, 90% nemenda gátu ekki forritað helmingunarleit rétt á nokkrum klukkustundum
- Könnun sýndi að helmingunarleit er bara rétt í 5 af hverjum 20 kennslubókum
- Jon Bentley skrifaði um þetta í bókinni *Programming Pearls* sem var gefin út 1986
- 20 árum seinna fannst villa í hans útfærslu á helmingunarleit
- Það var villa í innbyggðu Arrays.binarySearch helmingunarleitinni í Java þar til 2006