Složenost algoritama
10. predavanje

Saša Singer

singer@math.hr

web.math.hr/~singer

PMF – Matematički odjel, Zagreb

Hanojski tornjevi

Model — trajanje jednog poteza

Broj poteza za n diskova:

$$F(n) = 2^n - 1.$$

Ako je izmjereno vrijeme $T(n)$, onda je trajanje jednog poteza

$$c(n) = \frac{T(n)}{F(n)} = \frac{T(n)}{2^n - 1}.$$

Uočiti: $c(n)$ je obratno proporcionalan brzini (inverz brzine).

Hanoi — ispis na ekran

```
procedure Hanoi ( n, i, j : integer ) ;

begin

  if n > 0 then
    begin
    Hanoi ( n - 1, i, 6 - i - j ) ;
    writeln ( i, ' -> ', j ) ;
    Hanoi ( n - 1, 6 - i - j, j ) ;
    end ;  { n > 0 }

end ;  { Hanoi }
```

Hanoi — ispis na disk

```
procedure Hanoi ( n, i, j : integer ) ;

begin

  if n > 0 then
    begin
    Hanoi ( n - 1, i, 6 - i - j ) ;
    writeln ( Moves, i, ' -> ', j ) ;
    Hanoi ( n - 1, 6 - i - j, j ) ;
    end ;  { n > 0 }

end ;  { Hanoi }
```

Hanoi — varijanta 0

```
procedure Hanoi ( n, i, j : integer ) ;

begin

  if n > 0 then
    begin
    Hanoi ( n - 1, i, 6 - i - j ) ;
    Prebaci ( i, j ) ;
    Hanoi ( n - 1, 6 - i - j, j ) ;
    end ;  { n > 0 }

end ;  { Hanoi }
```

Hanoi — varijanta 1

```
procedure Hanoi ( n, i, j : integer ) ;

begin

  if n <= 1 then
    Prebaci ( i, j )
  else
    begin
    Hanoi ( n - 1, i, 6 - i - j ) ;
    Prebaci ( i, j ) ;
    Hanoi ( n - 1, 6 - i - j, j ) ;
    end ;  { n > 0 }

end ;  { Hanoi }
```

Hanoi — varijanta a0

```
procedure Hanoi ( n, i, j, k : integer ) ;

begin

  if n > 0 then
    begin
    Hanoi ( n - 1, i, k, j ) ;
    Prebaci ( i, j ) ;
    Hanoi ( n - 1, k, j, i ) ;
    end ;  { n > 0 }

end ;  { Hanoi }
```

Hanoi — varijanta a1

```
procedure Hanoi ( n, i, j, k : integer ) ;

begin

  if n <= 1 then
    Prebaci ( i, j )
  else
    begin
    Hanoi ( n - 1, i, k, j ) ;
    Prebaci ( i, j ) ;
    Hanoi ( n - 1, k, j, i ) ;
    end ;  { n > 0 }

end ;  { Hanoi }
```

Potprogram prebaci

Potprogram Prebaci samo zbraja (globalne) poteze:

```
procedure Prebaci ( i, j : integer ) ;

begin
    { Povecaj globalni brojac poteza.
      Trajanje je (skoro) konstantno:

          T(Prebaci) = c.}

  Broj_poteza := Broj_poteza + 1 ;

end ;  { Prebaci }
```

Tablica izmjerenih vremena $T(n)$

Usporedba izmjerenih vremena $T(n)$ (u s) za razna računala i razne varijante:

Varijanta	P120_166	Klamath	BabyBlue
ekran, $n = 15$	48.53	65.04	0.89
disk, $n = 15$	0.43	0.41	0.02
0, $n = 30$	453.34	193.76	19.12
1, $n = 30$	297.91	160.75	23.43
a0, $n = 30$	466.29	175.76	16.79
a1, $n = 30$	277.67	155.67	22.58

Grubo objašnjenje zelenih i crvenih rezultata za varijantu 1 (bez poziva za $n = 0$) ide malo kasnije!

Tablica trajanja poteza za $n = 30$

Usporedba trajanja jednog poteza (u s) za razna računala i razne varijante:

Varijanta	P120_166	Klamath	BabyBlue
ekran, $n = 15$	$1.48 \cdot 10^{-3}$	$1.98 \cdot 10^{-3}$	$2.70 \cdot 10^{-5}$
disk, $n = 15$	$1.30 \cdot 10^{-5}$	$1.25 \cdot 10^{-5}$	$6.11 \cdot 10^{-7}$
0, $n = 30$	$4.22 \cdot 10^{-7}$	$1.80 \cdot 10^{-7}$	$1.78 \cdot 10^{-8}$
1, $n = 30$	$2.77 \cdot 10^{-7}$	$1.50 \cdot 10^{-7}$	$2.18 \cdot 10^{-8}$
a0, $n = 30$	$4.34 \cdot 10^{-7}$	$1.64 \cdot 10^{-7}$	$1.56 \cdot 10^{-8}$
a1, $n = 30$	$2.59 \cdot 10^{-7}$	$1.45 \cdot 10^{-7}$	$2.10 \cdot 10^{-8}$

Objašnjenje rezultata za varijante 0 i 1

Do Pentium 4—Northwood procesora ponašanje je zeleno,

- isplati se izbaciti pozive za $n = 0$.

Od Pentium 4—Prescott procesora ponašanje je crveno,

- ne isplati se izbaciti pozive za $n = 0$.

Procesor ima "look–ahead" za instrukcije (u cacheu) i sam

- izbacuje ekstra pozive (instrukcije) koji ništa ne rade (ne mijenjaju ostale varijable).

Dakle, moderni procesori "misle" za nas. Nažalost, i stimuliraju loš stil programiranja!