Brza Fourierova transformacija

Filip Nikšić
fniksic@gmail.com

PMF – Matematički odjel

Složenost algoritama
4. lipnja 2007.

Pregled

Pregled

Pregled

Definicija problema

- Zadan je polinom

$$A(z) = a_0 + a_1 z + \ldots + a_{n-1} z^{n-1} .$$

- Izračunati vrijednosti

$$y_k = A(\omega_n^k), \quad k = 0, \ldots, n-1 ,$$

gdje su ω_n^k n-ti korijeni iz jedinice.

Diskretna Fourierova transformacija

Definicija

Diskretna Fourierova transformacija je preslikavanje $\mathrm{DFT}_n : a \mapsto y$ koje vektoru koeficijenata $a = (a_0, \ldots, a_{n-1})$ pridružuje vektor vrijednosti u n-tim korijenima iz jedinice $y = (y_0, \ldots, y_{n-1})$.

Vidjeli smo da je to linearno preslikavanje: $\mathrm{DFT}_n(a) = V_n a$, gdje je V_n Vandermondeova matrica.

Brza Fourierova transformacija

Vidjeli smo FFT algoritam koji iskorištava svojstva grupe n-tih korijena iz jedinice da izračuna $\mathrm{DFT}_n(a)$. Ideja:

1. Vektor a duljine n podijeli na dva vektora duljine $n/2$: $a^{[0]}$ s parnim i $a^{[1]}$ s neparnim indeksima.
2. Izračunaj $y^{[0]} = \mathrm{DFT}_{n/2}(a^{[0]})$ i $y^{[1]} = \mathrm{DFT}_{n/2}(a^{[1]})$.
3. Rekonstruiraj $y = \mathrm{DFT}_n(a)$ na sljedeći način:

$$y_k = y_k^{[0]} + \omega_n^k y_k^{[1]}$$
$$y_{k+n/2} = y_k^{[0]} - \omega_n^k y_k^{[1]} \qquad k = 0, \ldots, n/2 - 1$$

Brza Fourierova transformacija

Vidjeli smo FFT algoritam koji iskorištava svojstva grupe n-tih korijena iz jedinice da izračuna $\mathrm{DFT}_n(a)$. Ideja:

1. Vektor a duljine n podijeli na dva vektora duljine $n/2$: $a^{[0]}$ s parnim i $a^{[1]}$ s neparnim indeksima.
2. Izračunaj $y^{[0]} = \mathrm{DFT}_{n/2}(a^{[0]})$ i $y^{[1]} = \mathrm{DFT}_{n/2}(a^{[1]})$.
3. Rekonstruiraj $y = \mathrm{DFT}_n(a)$ na sljedeći način:

$$y_k = y_k^{[0]} + \omega_n^k y_k^{[1]}$$
$$y_{k+n/2} = y_k^{[0]} - \omega_n^k y_k^{[1]}$$
$$k = 0, \ldots, n/2 - 1$$

Brza Fourierova transformacija

Vidjeli smo FFT algoritam koji iskorištava svojstva grupe n-tih korijena iz jedinice da izračuna $\mathrm{DFT}_n(a)$. Ideja:

1. Vektor a duljine n podijeli na dva vektora duljine $n/2$: $a^{[0]}$ s parnim i $a^{[1]}$ s neparnim indeksima.
2. Izračunaj $y^{[0]} = \mathrm{DFT}_{n/2}(a^{[0]})$ i $y^{[1]} = \mathrm{DFT}_{n/2}(a^{[1]})$.
3. Rekonstruiraj $y = \mathrm{DFT}_n(a)$ na sljedeći način:

$$y_k = y_k^{[0]} + \omega_n^k y_k^{[1]} \qquad k = 0, \ldots, n/2 - 1$$
$$y_{k+n/2} = y_k^{[0]} - \omega_n^k y_k^{[1]}$$

Brza Fourierova transformacija

Vidjeli smo FFT algoritam koji iskorištava svojstva grupe n-tih korijena iz jedinice da izračuna $\mathrm{DFT}_n(a)$. Ideja:

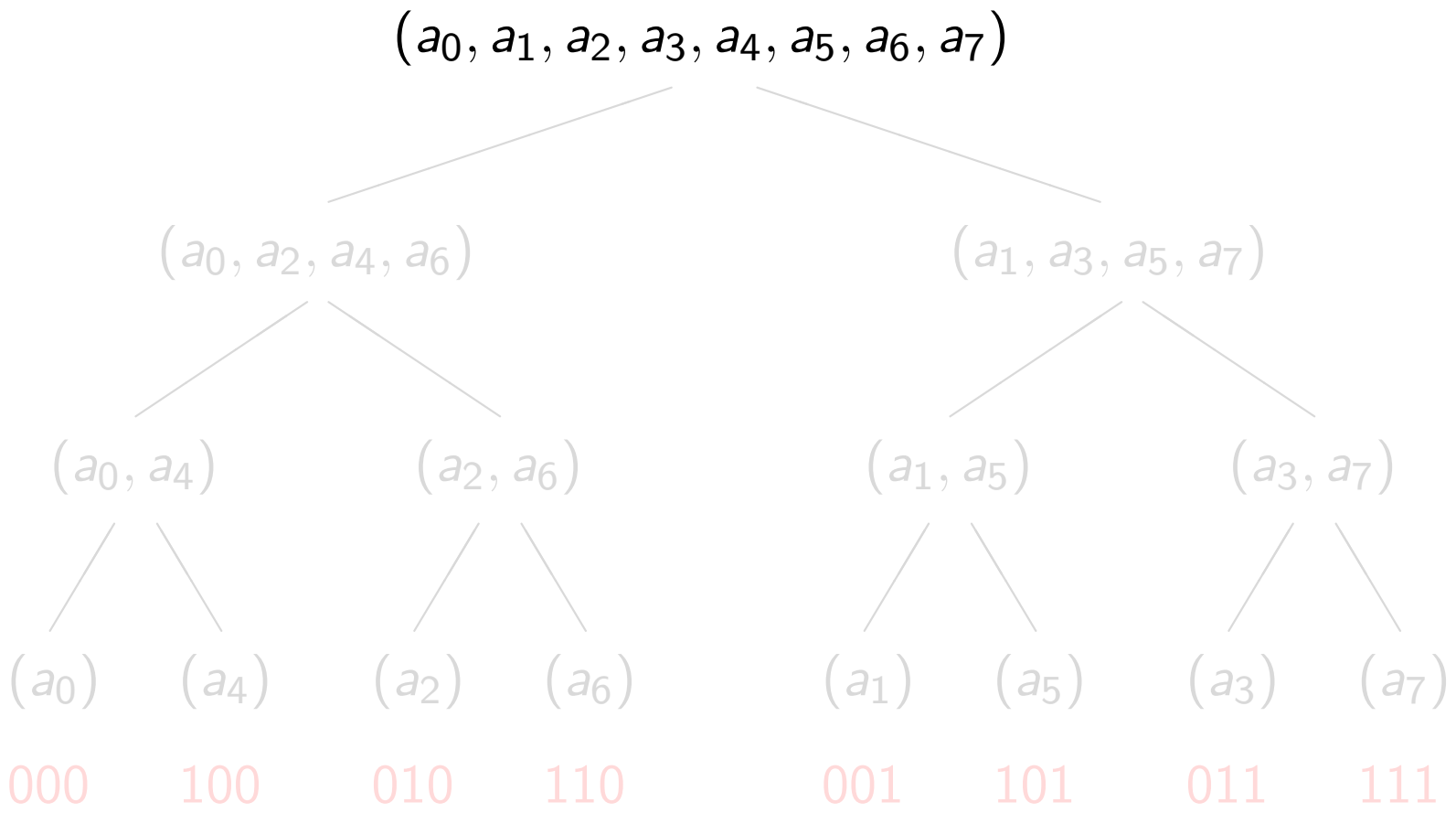
1. Vektor a duljine n podijeli na dva vektora duljine $n/2$: $a^{[0]}$ s parnim i $a^{[1]}$ s neparnim indeksima.
2. Izračunaj $y^{[0]} = \mathrm{DFT}_{n/2}(a^{[0]})$ i $y^{[1]} = \mathrm{DFT}_{n/2}(a^{[1]})$.
3. Rekonstruiraj $y = \mathrm{DFT}_n(a)$ na sljedeći način:

$$\begin{aligned} y_k &= y_k^{[0]} + \omega_n^k y_k^{[1]} \\ y_{k+n/2} &= y_k^{[0]} - \omega_n^k y_k^{[1]} \end{aligned} \qquad k = 0, \ldots, n/2 - 1$$

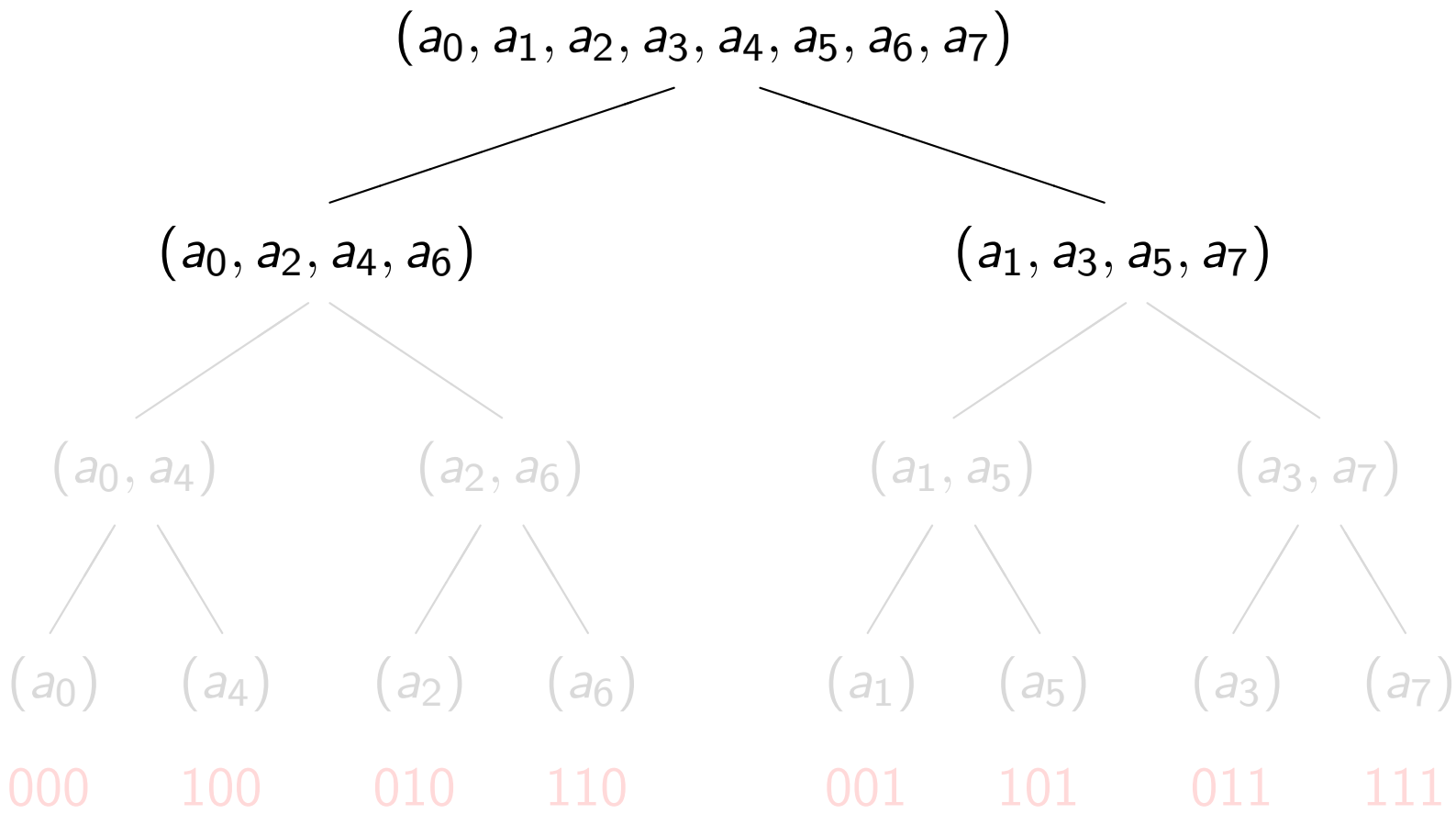
Svojstva rekurzivnog FFT-a

- Vremenska složenost je $\Theta(n \log n)$.
- Rekurzivan je.
- Duljina vektora mora biti 2^m.

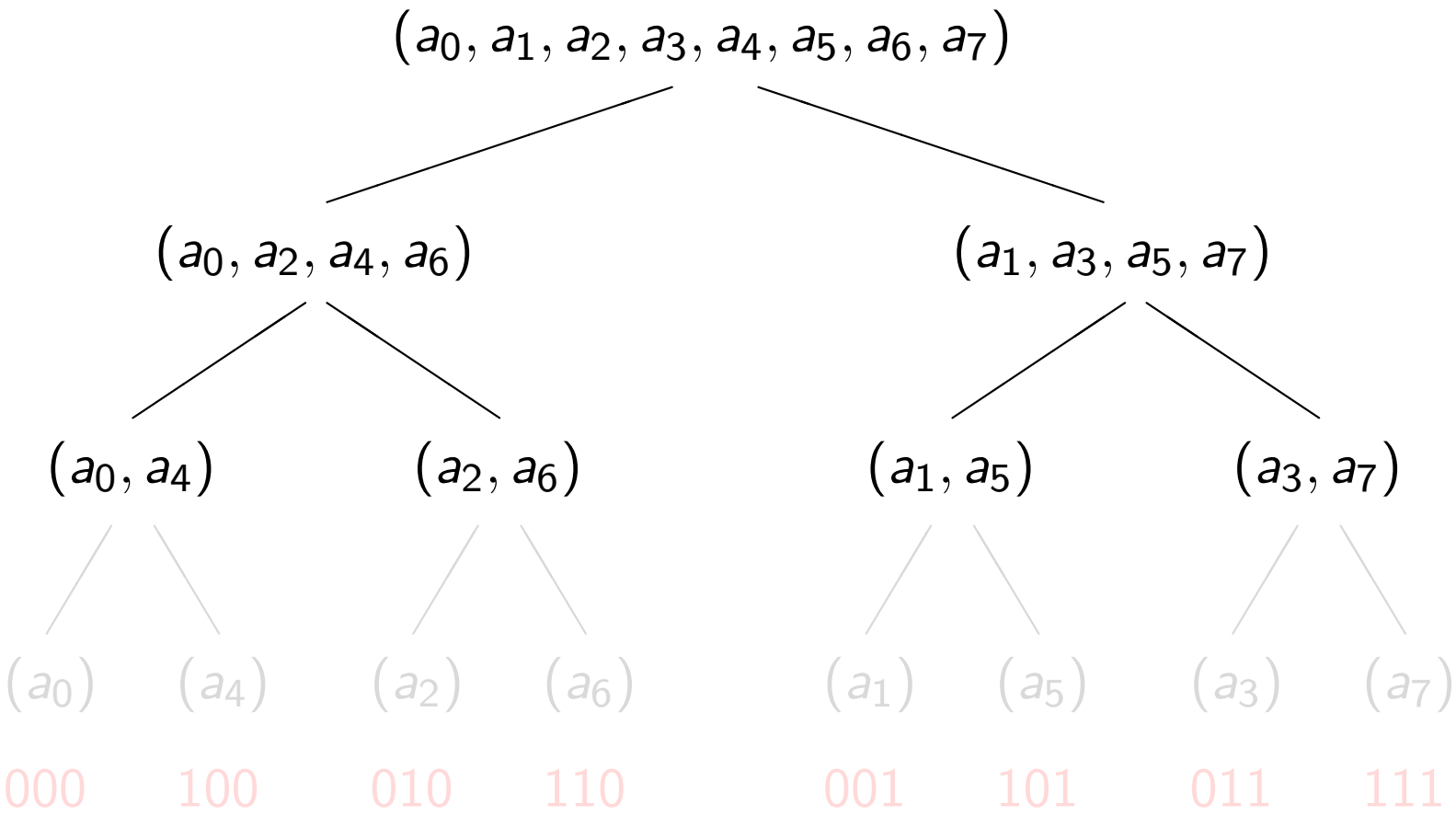
Stablo rekurzivnih poziva



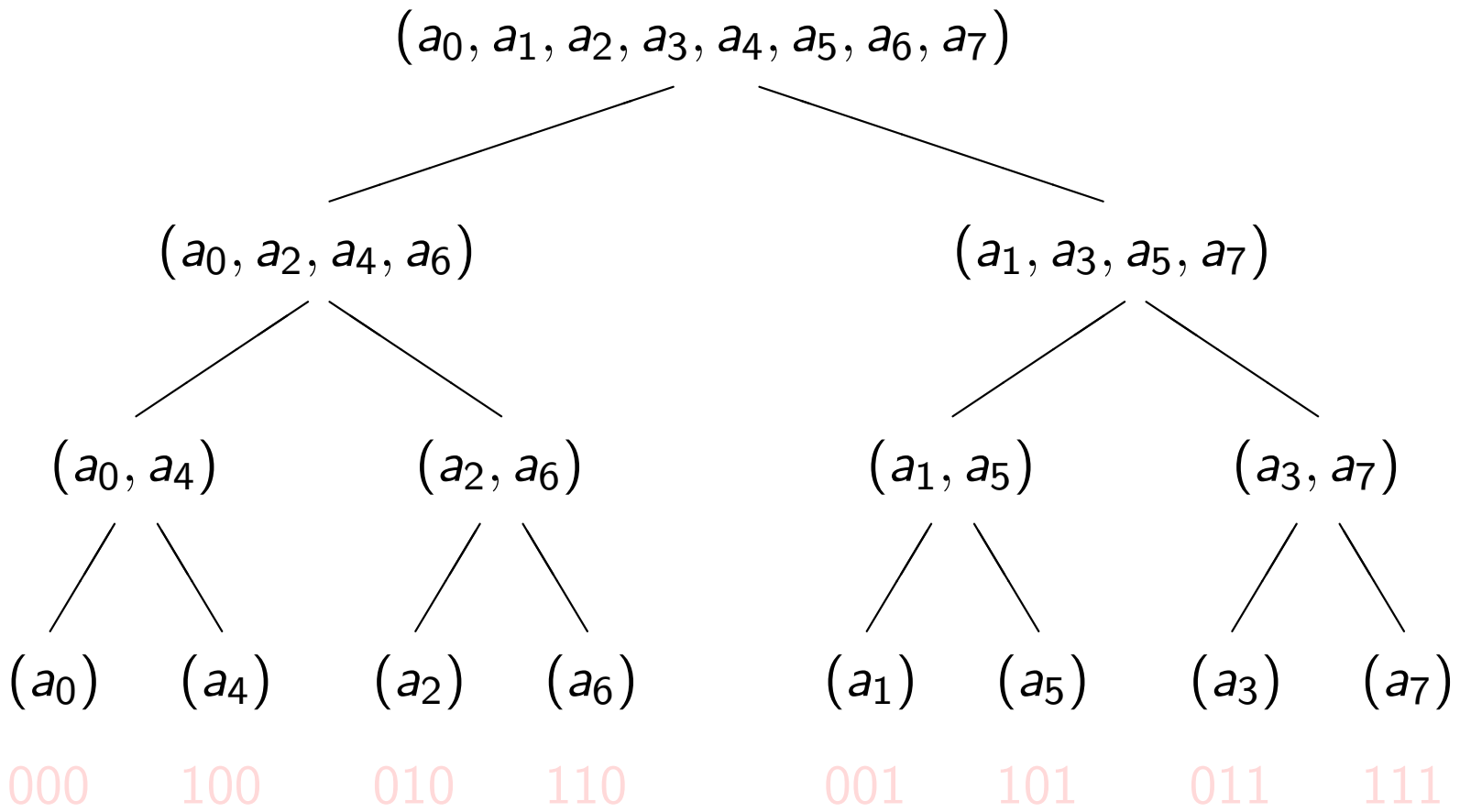
Stablo rekurzivnih poziva



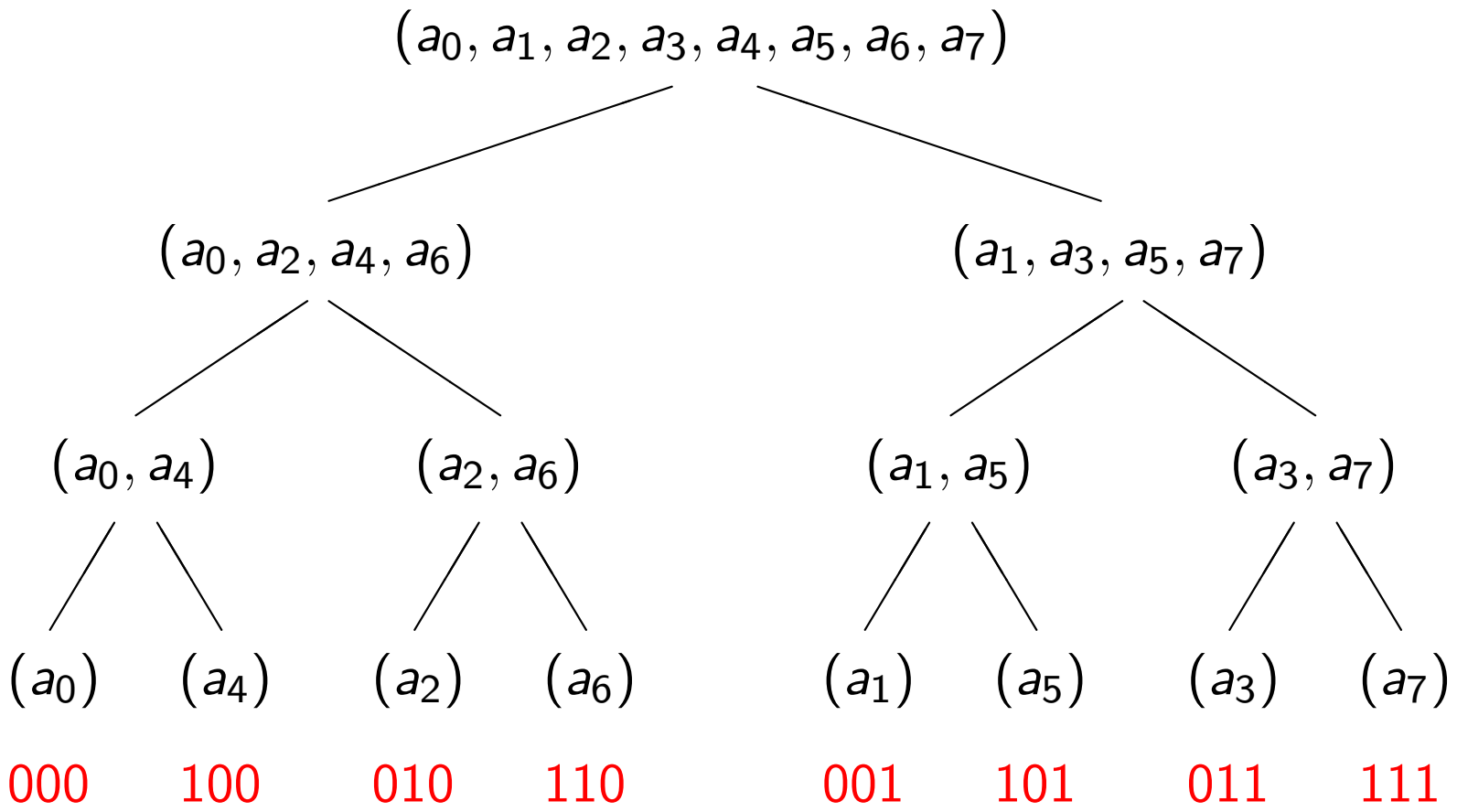
Stablo rekurzivnih poziva



Stablo rekurzivnih poziva



Stablo rekurzivnih poziva



Permutacija ulaznog vektora

- Iz vektora $a = (a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7)$ želimo dobiti $y = (a_0, a_4, a_2, a_6, a_1, a_5, a_3, a_7)$.
- Označimo s $\text{rev}(k)$ broj s obrnutim binarnim prikazom broja k. Primjerice,

$$\text{rev}(3) = \text{rev}(011_2) = 110_2 = 6\,.$$

- Tad je

$$y_k = a_{\text{rev}(k)}, \qquad k = 0, \ldots, n-1\,.$$

Permutacija ulaznog vektora

Implementacija

```
for (i=0, k=0; i<n; i++) {
    y[i]=a[k];

    /* Povecavamo "obrnuti" counter */
    for (j=1<<(lg(n)-1); k&j!=0; j>>=1) k^=j;
    k^=j;
}
```

Permutacija ulaznog vektora

Analiza složenosti

i	rev(i)
0	000
1	100
2	010
3	110
4	001
5	101
6	011
7	111

Primijetimo da se neki bitovi "flipaju" više puta. Npr. prvi bit slijeva se svaki put "flipa", a svaki sljedeći bit upola manje puta.
Ukupan broj "flipova" je

$$\sum_{k=0}^{\lg n-1} \frac{n}{2^k} \leqslant n \sum_{k=0}^{\infty} \frac{1}{2^k} = 2n\,.$$

Dakle, permutiranje ima složenost $\Theta(n)$.

Permutacija ulaznog vektora

Analiza složenosti

i	rev(i)
0	000
1	100
2	010
3	110
4	001
5	101
6	011
7	111

Primijetimo da se neki bitovi "flipaju" više puta. Npr. prvi bit slijeva se svaki put "flipa", a svaki sljedeći bit upola manje puta.
Ukupan broj "flipova" je

$$\sum_{k=0}^{\lg n-1} \frac{n}{2^k} \leqslant n \sum_{k=0}^{\infty} \frac{1}{2^k} = 2n\,.$$

Dakle, permutiranje ima složenost $\Theta(n)$.

Permutacija ulaznog vektora

Analiza složenosti

i	rev(i)
0	000
1	100
2	010
3	110
4	001
5	101
6	011
7	111

Primijetimo da se neki bitovi "flipaju" više puta. Npr. prvi bit slijeva se svaki put "flipa", a svaki sljedeći bit upola manje puta.
Ukupan broj "flipova" je

$$\sum_{k=0}^{\lg n-1} \frac{n}{2^k} \leqslant n \sum_{k=0}^{\infty} \frac{1}{2^k} = 2n\,.$$

Dakle, permutiranje ima složenost $\Theta(n)$.

Permutacija ulaznog vektora

Analiza složenosti

i	rev(i)
0	000
1	100
2	010
3	110
4	001
5	101
6	011
7	111

Primijetimo da se neki bitovi "flipaju" više puta. Npr. prvi bit slijeva se svaki put "flipa", a svaki sljedeći bit upola manje puta.
Ukupan broj "flipova" je

$$\sum_{k=0}^{\lg n-1} \frac{n}{2^k} \leqslant n \sum_{k=0}^{\infty} \frac{1}{2^k} = 2n\,.$$

Dakle, permutiranje ima složenost $\Theta(n)$.

Permutacija ulaznog vektora

Analiza složenosti

i	rev(i)
0	000
1	100
2	010
3	110
4	001
5	101
6	011
7	111

Primijetimo da se neki bitovi "flipaju" više puta. Npr. prvi bit slijeva se svaki put "flipa", a svaki sljedeći bit upola manje puta.
Ukupan broj "flipova" je

$$\sum_{k=0}^{\lg n-1} \frac{n}{2^k} \leqslant n \sum_{k=0}^{\infty} \frac{1}{2^k} = 2n\,.$$

Dakle, permutiranje ima složenost $\Theta(n)$.

Permutacija ulaznog vektora

Analiza složenosti

i	rev(i)
0	000
1	100
2	010
3	110
4	001
5	101
6	011
7	111

Primijetimo da se neki bitovi "flipaju" više puta. Npr. prvi bit slijeva se svaki put "flipa", a svaki sljedeći bit upola manje puta.
Ukupan broj "flipova" je

$$\sum_{k=0}^{\lg n-1} \frac{n}{2^k} \leqslant n \sum_{k=0}^{\infty} \frac{1}{2^k} = 2n\,.$$

Dakle, permutiranje ima složenost $\Theta(n)$.

Permutacija ulaznog vektora

Analiza složenosti

i	rev(i)
0	000
1	100
2	010
3	110
4	001
5	101
6	011
7	111

Primijetimo da se neki bitovi "flipaju" više puta. Npr. prvi bit slijeva se svaki put "flipa", a svaki sljedeći bit upola manje puta.
Ukupan broj "flipova" je

$$\sum_{k=0}^{\lg n-1} \frac{n}{2^k} \leqslant n \sum_{k=0}^{\infty} \frac{1}{2^k} = 2n\,.$$

Dakle, permutiranje ima složenost $\Theta(n)$.

Permutacija ulaznog vektora

Analiza složenosti

i	rev(i)
0	000
1	100
2	010
3	110
4	001
5	101
6	011
7	111

Primijetimo da se neki bitovi "flipaju" više puta. Npr. prvi bit slijeva se svaki put "flipa", a svaki sljedeći bit upola manje puta.
Ukupan broj "flipova" je

$$\sum_{k=0}^{\lg n-1} \frac{n}{2^k} \leqslant n \sum_{k=0}^{\infty} \frac{1}{2^k} = 2n\,.$$

Dakle, permutiranje ima složenost $\Theta(n)$.

Permutacija ulaznog vektora

Analiza složenosti

i	rev(i)
0	000
1	100
2	010
3	110
4	001
5	101
6	011
7	111

Primijetimo da se neki bitovi "flipaju" više puta. Npr. prvi bit slijeva se svaki put "flipa", a svaki sljedeći bit upola manje puta.
Ukupan broj "flipova" je

$$\sum_{k=0}^{\lg n-1} \frac{n}{2^k} \leqslant n \sum_{k=0}^{\infty} \frac{1}{2^k} = 2n\,.$$

Dakle, permutiranje ima složenost $\Theta(n)$.

Permutacija ulaznog vektora

Analiza složenosti

i	rev(i)
0	000
1	100
2	010
3	110
4	001
5	101
6	011
7	111

Primijetimo da se neki bitovi "flipaju" više puta. Npr. prvi bit slijeva se svaki put "flipa", a svaki sljedeći bit upola manje puta.
Ukupan broj "flipova" je

$$\sum_{k=0}^{\lg n-1} \frac{n}{2^k} \leqslant n \sum_{k=0}^{\infty} \frac{1}{2^k} = 2n\,.$$

Dakle, permutiranje ima složenost $\Theta(n)$.

Permutacija ulaznog vektora

Analiza složenosti

i	rev(i)
0	000
1	100
2	010
3	110
4	001
5	101
6	011
7	111

Primijetimo da se neki bitovi "flipaju" više puta. Npr. prvi bit slijeva se svaki put "flipa", a svaki sljedeći bit upola manje puta.
Ukupan broj "flipova" je

$$\sum_{k=0}^{\lg n-1} \frac{n}{2^k} \leqslant n \sum_{k=0}^{\infty} \frac{1}{2^k} = 2n\,.$$

Dakle, permutiranje ima složenost $\Theta(n)$.

Permutacija ulaznog vektora

Analiza složenosti

i	rev(i)
0	000
1	100
2	010
3	110
4	001
5	101
6	011
7	111

Primijetimo da se neki bitovi "flipaju" više puta. Npr. prvi bit slijeva se svaki put "flipa", a svaki sljedeći bit upola manje puta.
Ukupan broj "flipova" je

$$\sum_{k=0}^{\lg n-1} \frac{n}{2^k} \leqslant n \sum_{k=0}^{\infty} \frac{1}{2^k} = 2n\,.$$

Dakle, permutiranje ima složenost $\Theta(n)$.

Permutacija ulaznog vektora

Analiza složenosti

i	rev(i)
0	000
1	100
2	010
3	110
4	001
5	101
6	011
7	111

Primijetimo da se neki bitovi "flipaju" više puta. Npr. prvi bit slijeva se svaki put "flipa", a svaki sljedeći bit upola manje puta.
Ukupan broj "flipova" je

$$\sum_{k=0}^{\lg n-1} \frac{n}{2^k} \leqslant n \sum_{k=0}^{\infty} \frac{1}{2^k} = 2n\,.$$

Dakle, permutiranje ima složenost $\Theta(n)$.

Permutacija ulaznog vektora

Analiza složenosti

i	rev(i)
0	000
1	100
2	010
3	110
4	001
5	101
6	011
7	111

Primijetimo da se neki bitovi "flipaju" više puta. Npr. prvi bit slijeva se svaki put "flipa", a svaki sljedeći bit upola manje puta.
Ukupan broj "flipova" je

$$\sum_{k=0}^{\lg n-1} \frac{n}{2^k} \leqslant n \sum_{k=0}^{\infty} \frac{1}{2^k} = 2n\,.$$

Dakle, permutiranje ima složenost $\Theta(n)$.

Permutacija ulaznog vektora

Analiza složenosti

i	rev(i)
0	000
1	100
2	010
3	110
4	001
5	101
6	011
7	111

Primijetimo da se neki bitovi "flipaju" više puta. Npr. prvi bit slijeva se svaki put "flipa", a svaki sljedeći bit upola manje puta.
Ukupan broj "flipova" je

$$\sum_{k=0}^{\lg n-1} \frac{n}{2^k} \leqslant n \sum_{k=0}^{\infty} \frac{1}{2^k} = 2n\,.$$

Dakle, permutiranje ima složenost $\Theta(n)$.

Permutacija ulaznog vektora

Analiza složenosti

i	rev(i)
0	000
1	100
2	010
3	110
4	001
5	101
6	011
7	111

Primijetimo da se neki bitovi "flipaju" više puta. Npr. prvi bit slijeva se svaki put "flipa", a svaki sljedeći bit upola manje puta.
Ukupan broj "flipova" je

$$\sum_{k=0}^{\lg n-1} \frac{n}{2^k} \leqslant n \sum_{k=0}^{\infty} \frac{1}{2^k} = 2n\,.$$

Dakle, permutiranje ima složenost $\Theta(n)$.

Kôd

```
for (i=1; i<=lg(n); i++) {
    s=1<<i;
    ws.Re=cos(2*pi/s);
    ws.Im=sin(2*pi/s);
    w.Re=1.0;
    w.Im=0.0;
    for (j=0; j<s/2; j++) {
        for (k=j; k<n; k+=s) {
            t=w*y[k+s/2];
            u=y[k];
            y[k]=u+t;
            y[k+s/2]=u-t;
        }
        w=w*ws;
    }}
```

Složenost

Pažljivim brojanjem *realnih* floating-point operacija vidimo da ih je:

$$F(n) = 5n \lg n + 6n + 4 \lg n - 6, \quad n = 2^m, \ m \in \mathbb{N}_0 .$$

Mjerenje brzine

- Mjerenje brzine provedeno je na računalu s AMD Barton 2500+ (2.2 GHz) procesorom i 512 MB RAM memorije (266 MHz, dual-channel).
- Kompajler je GCC 4.1.1, prvo bez optimizacije, onda s opcijama -O3 -march=athlon-xp.
- "Brzina" je dobivena sljedećom formulom:

$$\text{mflops}(n) = F(n)/T(n)\,,$$

gdje je $F(n)$ broj realnih floating-point operacija, a $T(n)$ vrijeme u mikrosekundama.

Rezultati

Bez optimizacije

$lg(n)$	$T(n)$	mflops(n)
0	2	0.000000
1	8	2.500000
2	2	33.000000
3	2	87.000000
4	4	106.500000
5	6	167.666667
6	12	193.500000
7	26	202.692308
8	55	214.581818
9	131	199.557252
10	277	207.140794
11	604	206.897351

$lg(n)$	$T(n)$	mflops(n)
12	1318	205.142640
13	3180	182.917610
14	7765	160.364971
15	17738	149.637050
16	135225	41.679823
17	309778	38.503748
18	643567	39.103761
19	1385176	38.228469
20	3001342	37.033144
21	6513845	35.736796
22	14355966	33.891091

Rezultati

Optimizacija -O3 -march=athlon-xp

$lg(n)$	$T(n)$	mflops(n)
0	3	0.000000
1	8	2.500000
2	1	66.000000
3	1	174.000000
4	2	213.000000
5	3	335.333333
6	5	464.400000
7	8	658.750000
8	15	786.800000
9	40	653.550000
10	70	819.685714
11	174	718.195402

$lg(n)$	$T(n)$	mflops(n)
12	425	636.183529
13	2148	270.799814
14	5565	223.761725
15	14612	181.649466
16	125634	44.861693
17	297217	40.130995
18	632326	39.798917
19	1346117	39.337708
20	2928864	37.949570
21	6421021	36.253417
22	13832060	35.174757

Generalizacija leme raspolavljanja

Lema (Lema cijepanja)

Neka je $n = r \cdot s$. r-te potencije svih n-tih korijena jedinice su upravo svi s-ti korijeni iz jedinice. Preciznije:

$$((\omega_n^0)^r, (\omega_n^1)^r, \ldots, (\omega_n^{n-1})^r) = (\omega_s^0, \omega_s^1, \ldots, \omega_s^{s-1}, \omega_s^0, \omega_s^1, \ldots, \omega_s^{s-1})$$

Dokaz.

Direktno iz leme kraćenja. □

Generalizirani rastav polinoma

Polinom $A(z) = a_0 + a_1 z + \ldots a_{n-1} z^{n-1}$ rastavljamo ovako:

$$\begin{aligned} A(z) = a_0 + a_r z^r + \ldots + a_{(s-1)r} z^{(s-1)r} \\ + z(a_1 + a_{r+1} z^r + \ldots + a_{(s-1)r+1} z^{(s-1)r}) \\ \vdots \\ + z^{r-1}(a_{r-1} + a_{r+(r-1)} z^r + \ldots + a_{(s-1)r+(r-1)} z^{(s-1)r}) \end{aligned}$$

Generalizirani rastav polinoma

Uvođenjem polinoma

$$A^{[0]}(z) = a_0 + a_r z + \ldots + a_{(s-1)r} z^{s-1}$$
$$A^{[1]}(z) = a_1 + a_{r+1} z + \ldots + a_{(s-1)r+1} z^{s-1}$$
$$\vdots$$
$$A^{[r-1]}(z) = a_{r-1} + a_{r+(r-1)} + \ldots + a_{(s-1)r+(r-1)} z^{s-1}$$

možemo pisati

$$\begin{aligned} A(\omega_n^k) &= A^{[0]}\left((\omega_n^k)^r\right) + \omega_n^k A^{[1]}\left((\omega_n^k)^r\right) + \ldots + \omega_n^{k(r-1)} A^{[r-1]}\left((\omega_n^k)^r\right) \\ &= A^{[0]}(\omega_s^k) + \omega_n^k A^{[1]}(\omega_s^k) + \ldots + \omega_n^{k(r-1)} A^{[r-1]}(\omega_s^k) \end{aligned}$$

Rekurzivni algoritam

function FFT(a,n);
if n je prost **then**
  for $k = 0$ to $n - 1$ **do**
    $y_k := \sum_{j=0}^{n-1} a_k \omega_n^{kj}$;
  end for
else
  Neka je $n = r \cdot s$.
  for $i = 0$ to $r - 1$ **do**
    $a^{[i]} := (a_i, a_{r+i}, \ldots, a_{(s-1)r+i})$;
    $y^{[i]} := \mathrm{FFT}(a^{[i]}, s)$;
  end for
  for $k = 0$ to $n - 1$ **do**
    $y_k := \sum_{j=0}^{r-1} y^{[j]}_{k \bmod s} \omega_n^{kj}$;
  end for
end if
return y;

Složenost

Teorem

Neka je $n = p_1^{\alpha_1} \cdots p_k^{\alpha_k}$. Najbolji izbor faktorizacije $n = r \cdot s$ je da r bude jedan od prostih faktora. U tom slučaju, broj kompleksnih množenja je

$$g(n) = n(\alpha_1(p_1 - 1) + \alpha_2(p_2 - 1) + \cdots + \alpha_k(p_k - 1)).$$

Dokaz.

Vidi [3]. □


Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest.
Introduction to Algorithms.
The MIT Press, Cambridge, Massachusetts, 1990.


Wikipedia – `http://www.wikipedia.org/`.
Wikipedia, the free encyclopedia.


Herbert S. Wilf.
Algorithms and complexity.
Internet Edition, 1994.