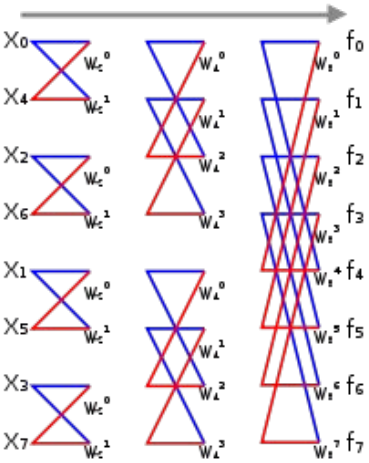

1 Algebarski algoritmi

1.1 Stepenovanje

Problem: Data su dva prirodna broja n i k. Izračunati n^k.

- TRIVIJALNI ALGORITAM: $n^k = n \cdot n^{k-1}$ zahteva k množenja.
- Ideja je da koristimo

$$n^k = \begin{cases} (n^{k/2})^2 & : k - parno \\ n \cdot (n^{(k-1)/2})^2 & : k - neparno \end{cases}$$

Algoritam 1 StepenKvadriradnjem(n, k)

Ulaz: n i k (dva prirodna broja)
Izlaz: P (vrednost izraza n^k)
begin
if $k = 1$ **then** $P := n$;
else z := StepenKvadriradnjem(n, k div 2);
  if $k \; mod \; 2 = 0$ **then** $P := z \cdot z$;
  else $P := n \cdot z \cdot z$;
  end if
end if
end

Broj množenja je $O(logk)$.

1. Okarakterisati vezu izmedju algoritma za računanje n^k kvadriranjem i binarne predstave broja k.

   Rešenje: Neka je $(k_{m-1}k_{m-2}\dots k_0)_2$ binarni zapis broja k. Rezultat inicijalno postavljamo na 1 ($z = 1$). Važi sledeće:

   $$k = (\dots(k_{m-1} \cdot 2 + k_{m-2}\dots) \cdot 2 + \dots) \cdot 2 + k_0$$

   Prolazimo bitove broja k počev od indeksa $m-1$, preko $m-2$...., do 0. Tekući rezultat kvadriramo, a ako je $k_i = 1$ onda množimo i sa brojem n. Dakle, $z := z^2 \cdot n^{k_i}$.

2. Algoritam *StepenKvadriradnjem* ne minimizira uvek broj množenja. Navesti primer izračunavanja n^k ($k > 10$) sa manjim brojem množenja nego u ovom algoritmu.

   Rešenje: Na primer, za stepenovanje eksponentom 15 algoritam *StepenKvadriradnjem* zahteva 6 množenja:

   $$((n^2 \cdot n)^2 \cdot n)^2 \cdot n = n^{15}$$

   Međutim stepen n^{15} se može izračunati i pomoću 5 množenja:

   $$((n^2)^2 \cdot n)^3 = n^{15}$$

1.2 Euklidov algoritam

Problem: Data su dva prirodna broja n i m. Odrediti njihov najveći zajednički delilac.

Algoritam 2 EuklidovAlgoritam(m,n)

Ulaz: m i n (dva prirodna broja)
Izlaz: NZD (najveći zajednički delilac brojeva m i n)
begin
$a := max(m, n)$;
$b := min(m, n)$;
$r := 1$;
while r> 0 **do**
    $r :=$ amod b;
    $a := b$;
    $b := r$;
end while
$NZD := a$;
end

1. Konstruisati algoritam za odredjivanje NZD (najvećeg zajedničkog delioca) k zadatih prirodnih brojeva.

   Rešenje: Neka su dati brojevi $a_1, a_2, \ldots, a_n$ i neka je a_1 najmanji među njima. NZD ovih brojeva se ne menja ako se brojevi $a_2, a_3, \ldots, a_n$ zamene svojim ostacima pri deljenju sa a_1 i zatim se iz skupa izbace sve nule. Ovim operacijama se smanjuje veličina najvećeg broja u skupu! Posle konačnog broja koraka dolazimo do sistema od samo jednog broja i to je NZD brojeva $a_1, a_2, \ldots, a_n$.

2. Konstruisati algoritam za odredjivanje NZS (najmanjeg zajedničkog sadržaoca) k zadatih prirodnih brojeva.

   Rešenje: Problem rešavamo indukcijom po k.

   - *Baza*: $k = 2$. Neka je $d = NZD(a, b)$. Onda je $a = a_1 \cdot d$, $b = b_1 \cdot d$, $(a_1, b_1) = 1$ $NZS(a, b) = a_1 \cdot b_1 \cdot d = \frac{ab}{NZD(a,b)}$ Dakle, određivanje NZS-a je svedeno na određivanje NZD-a.
   - *Induktivna hipoteza:* Pretpostavimo da smo izračunali $A_{k-1} = NZS(a_1, a_2, \ldots, a_{k-1})$.
     Onda je $A_k = NZS(a_1, a_2, \ldots, a_k) = NZS(A_{k-1}, a_k) = \frac{A_{k-1}a_k}{NZD(A_{k-1},a_k)}$
     Dakle, potrebno je još jednom izvršiti Euklidov algoritam.

3. Na bazi ideje Euklidovog algoritma napisati algoritam koji pronalazi bar jedno celobrojno rešenje jednačine $ax + by = NZD(a, b)$.

Rešenje: Pretpostavimo da je $a \geq b$. Na osnovu Euklidovog algoritma važi:

$$a = q_0 b + r_1, 0 \leq r_1 < b$$
$$b = q_1 r_1 + r_2, 0 \leq r_2 < r_1$$
$$r_1 = q_2 r_2 + r_3, 0 \leq r_3 < r_2$$
$$\vdots$$
$$r_{n-1} = q_n r_n + r_{n+1}, 0 \leq r_{n+1} < r_n$$
$$r_n = q_{n+1} r_{n+1}$$
$$r_{n+1} = NZD(a, b)$$

Ove jednakosti se mogu napisati i kao:

$$r_1 = a - q_0 b$$
$$r_2 = b - q_1 r_1 = b - q_1(a - q_0 b) = -q_1 a + (1 + q_0 q_1) b$$
$$r_3 = r_1 - q_2 r_2 = \dots$$
$$\vdots$$
$$r_{i+1} = r_{i-1} - q_i r_i = \dots = x_{i+1} a + y_{i+1} b$$
$$\vdots$$
$$r_{n+1} = x_{n+1} a + y_{n+1} b$$

za neke $x_i, y_i \in \mathcal{Z}$

Primetimo da je moguće izraziti $r_{n+1}(NZD(a, b))$ kao linearnu kombinaciju brojeva a i b i to u $n + 1$ koraka.

Ako je u i-tom koraku dobijeno:

$$r_i = x_i a + y_i b$$
$$r_{i-1} = x_{i-1} a + y_{i-1} b$$
$$r_{i+1} = r_{i-1} - q_i r_i = x_{i-1} a + y_{i-1} b - q_i(x_i a + y_i b)$$
$$= (x_{i-1} - q_i x_i) a + (y_{i-1} - q_i y_i) b = x_{i+1} a + y_{i+1} b$$

Dakle,

$$x_{i+1} = x_{i-1} - q_i x_i$$

$$y_{i+1} = y_{i-1} - q_i y_i$$

Početne vrednosti su: $x_{-1} = 1$, $x_0 = 0$, $y_{-1} = 0$ i $y_0 = 1$.

Zaključujemo da je za računanje tekućeg koeficijenta potrebno znati vrednosti prethodna dva odgovarajuća koeficijenta.

Algoritam 3 ModifikovaniEuklidovAlgoritam(m,n)

Ulaz: a i b (dva prirodna broja)
Izlaz: x i y (celi brojevi tako da važi da je $ax + by = NZD(a, b)$)
begin
uz_a_pred :=1;
uz_b_pred :=0;
uz_a :=0;
uz_b :=1;
while b<> 0 **do**
    $q :=$ $a$div b;
    $r :=$ $a$mod b;
    $a := b$;
    $b := r$;
    if $b <> 0$ **then**
        uz_a_temp:=uz_a;
        uz_b_temp :=uz_b;
        uz_a :=uz_a_pred - q·uz_a;
        uz_b :=uz_b_pred - q·uz_b;
        uz_a_pred :=uz_a_temp;
        uz_b_pred :=uz_b_temp;
    end if
    x := uz_a;
    y := uz_b;
end while
end

1.3 Množenje polinoma

Problem: Dati su polinomi $P = \sum_{i=0}^{n-1} p_i x^i$ i $Q = \sum_{i=0}^{n-1} q_i x^i$, stepena $n-1$. Zadatak je da se izračuna njihov proizvod.

Ukoliko PQ računamo direktnim množenjem članova, složenost će biti $O(n^2)$.

Poboljšanje:

$$P = P_1 + P_2 x^{n/2}$$

$$P_1 = p_0 + p_1 x + \cdots + p_{n/2-1} x^{n/2-1}$$

$$P_2 = p_{n/2} + p_{n/2-1} x + \cdots + p_{n-1} x^{n/2-1}$$

$$Q = Q_1 + Q_2 x^{n/2}$$

$$PQ = (P_1 + P_2 x^{n/2})(Q_1 + Q_2 x^{n/2}) = P_1 Q_1 + (P_1 Q_2 + P_2 Q_1) x^{n/2} + P_2 Q_2 x^n$$

Označimo:

$$A = P_1 Q_1$$

$$B = P_1 Q_2$$

$$C = P_2 Q_1$$

$$D = P_2 Q_2$$

Nisu nam potrebni B i C pojedinačno, već u formi $B + C$. Ako je $E = (P_1 + P_2)(Q_1 + Q_2)$, onda je $B + C = E - A - D$, pa je dovoljno izračunati tri proizvoda.

Složenost:

$$T(n) = 3T(n/2) + O(n) \implies T(n) = O(n^{log_2 3})$$

1. Izračunati proizvod polinoma $1 + 2x + 3x^2 + 4x^3$ i $4 - 3x + 2x^2 - x^3$ koristeći najviše devet množenja koeficijenata.

   Rešenje:

   $$P(x) = P_1(X) + x^2 P_2(x), P_1(x) = 1 + 2x, P_2(x) = 3 + 4x$$

   $$Q(x) = Q_1(X) + x^2 Q_2(x), Q_1(x) = 4 - 3x, Q_2(x) = 2 - x$$

   $$P(x)Q(x) = (1 + 2x + (3 + 4x)x^2)(4 - 3x + (2 - x)x^2)$$

$$= (1+2x)(4-3x) + ((1+2x+3+4x)(4-3x+2-x)$$
$$-(1+2x)(4-3x) - (3+4x)(2-x))x^2 + (3+4x)(2-x)x^4$$
$$= (1+2x)(4-3x) + ((4+6x)(6-4x) - (1+2x)(4-3x)$$
$$-(3+4x)(2-x))x^2 + (3+4x)(2-x)x^4$$

Ovde imamo 3 različita množenja. Svako od njih predstavimo na sledeći način:

$$(1+2x)(4-3x) = 1 \cdot 4 + x(3 \cdot 1 - 1 \cdot 4 + 2 \cdot 3) + x^2 \cdot 2 \cdot 3$$

Tri množenja: $1 \cdot 4$, $3 \cdot 1$, $2 \cdot 3$.

$$(4+6x)(6-4x) = 4 \cdot 6 + x(10 \cdot 2 - 4 \cdot 6 + 6 \cdot 4) + x^2 \cdot 6 \cdot 4$$

Tri množenja: $4 \cdot 6$, $10 \cdot 2$, $6 \cdot 4$.

$$(3+4x)(2-x) = 3 \cdot 2 + x(7 \cdot 1 - 3 \cdot 2 + 4 \cdot 1) + x^2 \cdot 4 \cdot 1$$

Tri množenja: $3 \cdot 2$, $7 \cdot 1$, $4 \cdot 1$.

Dakle, ukupan broj množenja je 9.

2. Kako se mogu pomnožiti dva kompleksna broja $a+bi$ i $c+di$ pomoću samo tri množenja?

   Rešenje: Dovoljno je izračunati proizvode $p_1 = ac$, $p_2 = bd$, $p_3 = (a+b)(c+d) = p_1 + p_2 + (ad+bc)$.

$$(a+bi)(c+di) = ac - bd + i(ad+bc) = p_1 - p_2 + i(p_3 - p_1 - p_2)$$

3. Neka su A i B kvadratne matrice reda n čije su vrste otvoreni Grejovi kodovi (elementi su im iz skupa $0, 1$, a svake dve uzastopne vrste se razlikuju na tačno jednom mestu). Konstruisati algoritam složenosti $O(n^2)$ za množenje ovakvih matrica.

   Rešenje: $A = (a_{ij})$, $B = (b_{ij})$, $C = (c_{ij})$

   Biće nam doboljno to što su vrste matrice A otvoreni Grejovi kodovi.

   Računamo prvu vrstu matrice C:

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

$O(n^2)$ operacija za sve elemente prve vrste.

Konstruišemo vektor R tako da je $R[i]$ pozivija na kojoj se razlikuju i-ta i $(i+1)$-va vrsta matrice A, $i = 1, 2, \ldots, n-1$. Za ovo je takođe potrebno $O(n^2)$ operacija.

Važi:

$$c_{i+1,j} - c_{ij} = \sum_{k=1}^{n} a_{i+1,k} b_{kj} - \sum_{k=1}^{n} a_{ik} b_{kj}$$

$$= \sum_{k=1}^{n} (a_{i+1,k} - a_{ik}) b_{kj} = (a_{i+1,R[i]} - a_{i,R[i]}) b_{R[i]j}$$

Svi ostali elementi se računaju korigovanjem vrednosti iznad sebe (1 sabiranje, 1 oduzimanje, 1 množenje).

Dakle, ukupno $O(n^2)$ operacija.

2 Numerički algoritmi - FFT i množenje polinoma

Problem: Dati su polinomi $p(x)$ i $q(x)$. Zadatak je da se izračuna njihov proizvod.

Polinom stepena $n-1$ se, osim nizom svojih koeficijenata, može predstaviti i vrednostima u n različitih tačaka. Proizvod dva polinoma stepena $n-1$ je polinom stepena $2n-2$, pa ako su date vrednosti činioca u $2n-1$ tačaka, proizvod se može izračunati uz pomoć $2n-1$ množenja, tj.. u $O(n^2)$.

Prelaz od predstave koeficijentima na predstavu vrednostima u tačkama rešava se izračunavanjem vrednosti polinoma (npr. pomoću Horneove sheme). Dakle, računanje vrednosti polinoma $p(x)$ u n tačaka izvoljivo je pomoću n^2 množenja.

Prelaz od predstave vrednostima na predstavu koeficijentima se naziva interpolacija i u opštem slučaju zahteva $O(n^2)$ operacija.

FFT pogodnim izborom skupa tačaka uspeva da efikasnije izvrši obe operacije - algoritam je složenosti $O(nlogn)$. Dobar izbor je

$$x_{n/2+j} = -x_j$$

Ako je $P = \sum_{i=0}^{n-1} a_i x^i$, važi:

$$P(x) = P_e(x^2) + xP_o(x^2)$$

$$P_e(x) = \sum_{j=0}^{n/2-1} a_{2j}x^j, P_o(x) = \sum_{j=0}^{n/2-1} a_{2j+1}x^j$$

$$P(-x) = P_e(x^2) + (-x)P_o(x^2)$$

$$\vdots$$

Dobili smo dva potproblema dimenzije $n/2 + O(n)$ dopunskih operacija.

Dobija se da treba uzeti n-ti koren iz jedinice w:

$w^n = 1, w^j \neq 1$ za $0 < j < n$ Za n tačaka biramo $1, w, w^3, \ldots, w^{n-1}$.

Algoritam 4 $FFT(n, a_0, a_1, \ldots, a_{n-1}, w, V)$

Ulaz: n (prirodan broj, stepen dvojke) $a_0, a_1, \ldots, a_{n-1}$ niz elemenata w - primitivni n-ti koren iz jedinice
Izlaz: V(niz izlaznih elemenata sa indeksima od 0 i $n-1$)
begin
if $n = 1$ **then** $V[0] := a_0$
else
    $FFT(n/2, a_0, a_2, \ldots, a_{n-2}, w^2, U)$;
    $FFT(n/2, a_1, a_3, \ldots, a_{n-1}, w^2, W)$;
    for $j := 0$ to $n/2 - 1$ **do**
        $V[j] := U[j] + w^j W[j]$;
        $V[j + n/2] := U[j] - w^j W[j]$;
    end for
end if
end

Pokazuje se da je problem interpolacije veoma sličan problemu izračunavanja vrednosti, i da se praktično može iskoristiti isti algoritam.

Ako je

$$V(w) = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & w & \cdots & w^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & w^{n-1} & \cdots & w^{(n-1)(n-1)} \end{pmatrix}$$

$$a = \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix}$$

$$w = \begin{pmatrix} P(1) \\ P(w) \\ \vdots \\ P(w^{n-1}) \end{pmatrix}$$

Onda važi $V(w)a = v$

$$V^{-1}(w)V(w)a = V^{-1}(w)v$$

$$a = V^{-1}(w)v = 1/nV(w^{-1})v$$

I ovo se računa pomoću FFT - tako što se w zameni sa w^{-1}. Ovo je *inverzna Furijeova transformacija.*

1. Izračunati brzu Furijeovu transformaciju vektora $P(x) = 1+2x+3x^2+4x^3$.

   Rešenje: Dakle, dat je polinom sa koeficijentima $(1, 2, 3, 4)$. Potprobleme ćemo označavati sa $P_{j_0,\ldots,j_k}(x_0, \ldots, x_k)$ gde $j_0, \ldots, j_k$ označavaju koeficijente polinoma, a $x_0, \ldots, x_k$ tačke u kojima se izračunavaju vrednosti polinoma.

   Treba da rešimo $P_{1,2,3,4}(1, w, w^2, w^3)$, pri čemu važi $w^4 = 1$ i $w^2 = -1$.

   U prvom koraku problem svodimo na $P_{1,3}(1, w^2)$ i $P_{2,4}(1, w^2)$, a korišćenjem veze

   $$P(x) = P_e(x^2) + xP_o(x^2)$$

   dobijamo:

   $$P_{1,3}(1) = P_1(1^2) + 1 \cdot P_3(1^2) = P_1(1) + P_3(1) = 1 + 3 = 4$$
   $$P_{1,3}(w^2) = P_1(w^4) + w^2 \cdot P_3(w^4) = P_1(1) - P_3(1) = 1 - 3 = -2$$
   $$\implies P_{1,3}(1, w^2) = (4, -2)$$

   $$P_{2,4}(1) = P_2(1^2) + 1 \cdot P_4(1^2) = P_2(1) + P_4(1) = 2 + 4 = 6$$
   $$P_{2,4}(w^2) = P_2(w^4) + w^2 \cdot P_4(w^4) = P_2(1) - P_4(1) = 2 - 4 = -2$$
   $$\implies P_{2,4}(1, w^2) = (6, -2)$$

   $$P_{1,2,3,4}(1) = P_{1,3}(1^2) + 1 \cdot P_{2,4}(1^2) = 4 + 6 = 10$$
   $$P_{1,2,3,4}(w) = P_{1,3}(w^2) + w \cdot P_{2,4}(w^2) = -2 + w(-2) = -2 - 2w$$
   $$P_{1,2,3,4}(w^2) = P_{1,3}(w^4) + w^2 \cdot P_{2,4}(w^4) = P_{1,3}(1) - P_{2,4}(1) = 4 - 6 = -2$$
   $$P_{1,2,3,4}(w^3) = P_{1,3}(w^6) + w^3 \cdot P_{2,4}(w^6) = P_{1,3}(w^2) - wP_{2,4}(w^2)$$
   $$= -2 - w(-2) = -2 + 2w$$

   Dakle,

   $$P_{1,2,3,4}(1, w, w^2, w^3) = (10, -2 - 2w, -2, -2 + 2w)$$

2. Izračunati inverznu Furijeovu transformaciju vektora $(10, -2-2w, -2, -2+2w)$.

Rešenje: Zadatak je da rešimo $P_{10,-2-2w,-2,-2+2w}(1, w^{-1}, w^{-2}, w^{-3})$ i na kraju sve pomnožimo sa $\frac{1}{n} = \frac{1}{4}$.

$$P_{10,-2}(1) = P_{10}(1^2) + 1 \cdot P_{-2}(1^2) = 10 - 2 = 8$$
$$P_{10,-2}(w^{-2}) = P_{10}(w^{-4}) + w^{-2} \cdot P_{-2}(w^{-4}) = 10 - 1(-2) = 12$$
$$\implies P_{10,-2}(1, w^{-2}) = (8, 12)$$

$$P_{-2-2w,-2+2w}(1) = P_{-2-2w}(1^2)+1\cdot P_{-2+2w}(1^2) = -2-2w-2+2w = -4$$
$$P_{-2-2w,-2+2w}(w^{-2}) = P_{-2-2w}(w^{-4})+w^{-2}\cdot P_{-2+2w}(w^{-4}) = -2-2s+2-2w = -4w$$
$$\implies P_{-2-2w,-2+2w}(1, w^{-2}) = (-4, -4w)$$

$$P_{10,-2-2w,-2,-2+2w}(1, w^{-1}, w^{-2}, w^{-3}) = ?$$

$$P_{10,-2-2w,-2,-2+2w}(1) = P_{10,-2}(1^2) + 1 \cdot P_{-2-2w,-2+2w}(1^2) = 8 - 4 = 4$$
$$P_{10,-2-2w,-2,-2+2w}(w^{-1}) = P_{10,-2}(w^{-2}) + w^{-1} \cdot P_{-2-2w,-2+2w}(w^{-1}) =$$
$$= 12 + w^{-1}(-4w) = 12 - 4 = 8$$
$$P_{10,-2-2w,-2,-2+2w}(w^{-2}) = P_{10,-2}(w^{-4})+w^{-2}\cdot P_{-2-2w,-2+2w}w^{-4}) = 8-(-4) = 12$$
$$P_{10,-2-2w,-2,-2+2w}(w^{-3}) = P_{10,-2}(w^{-6})+w^{-3}\cdot P_{-2-2w,-2+2w}(w^{-3}) = 12+4 = 16$$
$$\{w^{-4} = 1, w^{-2} = -1\}$$

$$\implies P_{10,-2-2w,-2,-2+2w}(1, w^{-1}, w^{-2}, w^{-3}) = (4, 8, 12, 16)$$

Kada sve pomnožimo sa $\frac{1}{4}$ dobijamo $(1, 2, 3, 4)$.