Konstrukcija algoritma za DFT$_n$, uz opći $n \in \mathbb{N}$.

Ključni prvi korak u konstrukciji brzog algoritma za $n = 2^m$ je mogućnost rastava polinoma A na dva dijela iste duljine

$$n = 2 \cdot \left(\frac{n}{2}\right).$$

Ako je $n = p$ prost broj, onda takva mogućnost ne postoji. Tada je

$$y_k = A(\omega_p^k) = \sum_{j=0}^{p-1} a_j \omega_p^{kj}, \quad k = 0, \dots, p-1.$$

Znamo da je

$$\omega_p^{k \cdot j} = \omega_p^{k \cdot j \bmod p},$$

ali za prost broj p, eksponenti $k \cdot j \bmod p$ za $j = 0, \dots, p-1$ prolaze svih p vrijednosti

$$0, 1, \dots, p-1$$

za svaki $k \neq 0$. Naime, ω_p^0 je neutral u multiplikativnoj grupi p-tih korijena iz jedinice, a svih ostalih $p-1$ elemenata su generatori te grupe $\Leftrightarrow$ njihovim potenciranjem možemo dobiti sve elemente grupe.

Dakle, jedini direktni račun vektora y ide:

(a) korištenjem Hornerove sheme za polinom A stupnja $p-1$ (reda p), što treba

$$p-1\,M, p-1\,A \quad (\text{nad } \mathbb{C})$$

po svakoj točki ω_p^k (osim za ω_p^0, gdje množenja možemo ignorirati, ali moguću uštedu za jednu točku smo mogli napraviti i ranije; a nismo ju uzeli u obzir – ionako je nižeg reda veličine od ukupne složenosti).

(b) mogli bismo unaprijed spremiti (kao i prije) cijelu tablicu potencija

$$\omega_p^0, \ldots, \omega_p^{p-1}$$

pa imamo skalarni produkt vektora duljine p po svakoj točki - što je ukupno

$$p-1\ M,\ p-1\ A$$

operacija nad $\mathbb{C}$.

Dakle, u oba slučaja, za svih p točaka imamo

$$p(p-1)\ M,\ p(p-1)\ A \quad \text{za } DFT_p(a).$$

(množenja bismo mogli smanjiti na $(p-1)\cdot(p-1)$, ali zanemarimo to.).

- Finalno skaliranje za $DFT_p^{-1}(a)$ isto nećemo ovdje brojati!

Vidimo da ako je $n=p$ prost broj, onda <u>ne postoji</u> direktna <u>brza</u> diskretna Fourierova transformacija za a.

(Kasnije ćemo pokazati da se ipak i $DFT_p(a)$ može naći brzo, u $O(p \log p)\,M$, ali ne direktno, nego transformacijom vektora a i korištenjem konvolucije - kao kod brzog množenja polinoma).

- Pretpostavimo sad da je n složen broj

$$n = n_1 \cdot n_2, \quad n_1, n_2 > 1.$$

Ovdje su n_1 i n_2 bilo koji faktori, ne nužno prosti brojevi!

Tada A možemo rastaviti u n_1 "blokova", svaki duljine n_2.

Indekse ima $[\emptyset], [1]$ za $n_1 = 2$, sada opet odgovaraju ostaci modulo n_1, tj.

$$l = [\emptyset], [1], \ldots, [n_1 - 1].$$

(ako idemo kao u prvom DFT_{2^m} algoritmu)

$j \to \ell$, $k \to m$
$p \to n_1$, $q \to n_2$

[Odmah rastavi: $j = \ell + m \cdot n_1$, $\ell = j \bmod n_1$
$k = r + s \cdot n_2$, $r = k \bmod n_2$]

Henrici 3 ↗

Niz $a^{[\ell]}$ sadrži one elemente a_j od a koji imaju ostatak ℓ mod n_1. Dakle

$$a^{[\ell]} = (a_\ell, a_{\ell+n_1}, \dots, a_{\ell+(n_2-1)\cdot n_1}), \quad \ell = 0, \dots, n_1 - 1.$$

Pripadni polinomi su

$$A^{[\ell]}(x) = a_\ell + a_{\ell+n_1} x + \dots + a_{\ell+(n_2-1)\cdot n_1} x^{n_2-1}.$$

Odatle treba uvrstiti x^{n_1} i napraviti pravu linearnu kombinaciju:

$$A(x) = \sum_{j=0}^{n-1} a_j x^j = \left\{ j = \ell + m \cdot n_1, \ \begin{matrix} \ell = 0, \dots, n_1 - 1 \\ m = 0, \dots, n_2 - 1 \end{matrix} \right\}$$

$$= \sum_{m=0}^{n_2-1} \sum_{\ell=0}^{n_1-1} a_{\ell+m\cdot n_2} x^{\ell+m\cdot n_1}$$

$$= \sum_{\ell=0}^{n_1-1} x^\ell \cdot \sum_{m=0}^{n_2-1} a_{\ell+m\cdot n_2} (x^{n_1})^m$$

$$= \sum_{\ell=0}^{n_1-1} x^\ell \cdot A^{[\ell]}(x^{n_1}).$$

Sad uvrstimo $x = \omega_n^k$ i iskoristimo

$$\omega_{n_1 \cdot n_2}^{k \cdot n_1} = \omega_{n_2}^k \qquad (\text{za sve } k = 0, \dots, n-1)$$

onda je

$$y_k = \sum_{\ell=0}^{n_1-1} \omega_n^{\ell \cdot k} \cdot A^{[\ell]}(\omega_n^{k \cdot n_1}) \quad \text{→ sa koristimo prethodnu formulu } \omega_{n_1 \cdot n_2}^{k \cdot n_1} = \omega_{n_2}^k$$

$$= \sum_{\ell=0}^{n_1-1} \omega_n^{\ell \cdot k} \cdot A^{[\ell]}(\omega_{n_2}^k)$$

$$= \omega_{n_2}^{k \bmod n_2} \quad (\text{iz periodičnosti potencija od } \omega_{n_2})$$

$$= \sum_{\ell=0}^{n_1-1} \omega_n^{\ell \cdot k} \cdot A^{[\ell]}(\omega_{n_2}^{k \bmod n_2})$$

Neka je $y^{[e]} = DFT_{n_2}(a^{[e]})$, $e = 0, -, n_1 - 1$

To znači da je

$$y^{[e]}_{k \bmod n_2} = A^{[e]}(\omega_{n_2}^{k \bmod n_2}), \quad \forall k$$

($k \bmod n_2$ uredno prolazi $0, -, n_2 - 1$ i to n_1 puta, kad k ide od $\emptyset$ do $n-1$).

Dakle, dobivamo da je

$$y_k = \sum_{e=0}^{n_1-1} \omega_n^{e \cdot k} \cdot y^{[e]}_{k \bmod n_2}, \quad k = 0, -, n-1$$

Ako svaki y_k računamo po ovoj formuli, uz pretpostavku da su $\omega_n^{e \cdot k}$ tabelirani, imamo n skalarnih produkata vektora duljine n_2. Čak malo bolje, jer znamo da je

$$\omega_n^{e \cdot k} = 1 \quad \text{za } e = 0,$$

pa ova formula ima oblik

$$y_k = y^{[\emptyset]}_{k \bmod n_2} + \sum_{e=1}^{n_1-1} \omega_n^{e \cdot k} \, y^{[e]}_{k \bmod n_2}, \quad k = 0, -, n-1$$

---

dalje, trebamo

$n_1 - 1$ M, $n_1 - 1$ A po svakom y_k

ili ukupno

$n \cdot (n_1 - 1)$ M, $n \cdot (n_1 - 1)$ A (nad $\mathbb{C}$)

operacija da iz vektora $y^{[e]}$, $e = 0, -, n_1 - 1$ izračunamo traženi y.

Vidimo da u ovoj formulaciji imamo isti broj množenja M i zbrajanja A (nad $\mathbb{C}$), pa možemo analizirati samo množenja.

Što smo zapravo napravili?

Koristeći rastav $n = n_1 \cdot n_2$, svели smo računanje diskretne Fourierove transformacije duljine n, tj. DFT_n, na

1. n_1 računanja DFT_{n_2}, da izračunamo:

$$y^{[\ell]} = DFT_{n_2}(a^{[\ell]}), \quad \ell = 0, -, n_1 - 1$$

2. petlju za računanje $y = DFT_n(a)$ iz $y^{[\ell]}$ u obliku:

$$y_k = y^{[0]}_{k \bmod n_2} + \sum_{\ell=1}^{n_1-1} \omega_n^{\ell \cdot k} \cdot y^{[\ell]}_{k \bmod n_2}, \quad k = 0, -, n-1.$$

Ova zadnja relacija upravo nije ništa drugo nego Hornerova shema za

$$A(x) = \sum_{\ell=0}^{n_1-1} A^{[\ell]}(x^{n_1}) \cdot x^{\ell}$$

u točkama $x = \omega_n^k$, $k = 0, -, n-1$, tj. skalarni produkt vektora duljine n_1, s tim da u prvom sumandu ($\ell = 0$), nema množenja.

Ako s $M(n)$ označimo broj kompleksnih množenja za računanje DFT_n, onda po ovom algoritmu trebamo

$$M(n) = \underbrace{n \cdot (n_1 - 1)}_{\text{faza 2 - Horner}} + \underbrace{n_1 \cdot M(n_2)}_{\text{faza 1 - } n_1 \times DFT_{n_2}},$$

s tim da je $n = n_1 \cdot n_2$.

Prvo uočimo da algoritam i prethodna formula vrijede i za slučaj da je $n = p$ prost broj. Tada znamo da je

$$(\min)\ M(p) = p \cdot (p-1).$$

No, znamo i to da je

$$(\min)\ M(1) = 0,$$

jer se računanje $y = DFT_1(a)$ svodi na kopiranje $y = a$ (tj. $y_0 = a_0$), pa nema aritmetičkih operacija.

Ako $n=p$ "faktoriziramo" u obliku $p = 1 \cdot p$, tj. uzmemo $n_1 = 1$, $n_2 = p$, onda u fazi 1 imamo jedan DFT_p (što je ekvivalentno polaznom problemu), a faza 2 se svodi na n kopiranja $y_k = y_k^{[\emptyset]}$ (jer je $k \bmod n = k \bmod p = k$). Dakle:

$$M(p) = \underbrace{p \cdot (1-1)}_{\emptyset} + 1 \cdot M(p) = M(p)$$

što je besmisleno, ali pokazuje da je relacija za $M(n)$ konzistentna s $n_1 = 1$ – ne samo za $n=p$ već i općenito:

$$n = 1 \cdot n \qquad M(n) = \underbrace{n \cdot (1-1)}_{\emptyset} + 1 \cdot M(n) = M(n).$$

- S druge strane, ako pišemo $p = p \cdot 1$, onda u fazi 1 imamo p "računanja" DFT_1, što je p kopiranja, bez aritmetičkih operacija, a u fazi 2 imamo Hornerovu shemu za polinom reda p (stupnja $p-1$) i to p puta, što bismo i inače iskoristili za DFT_p.

Dakle, u ovim rastavima smijemo uzeti (općenito) $n_2 = 1$, pa se algoritam svodi na običnu Hornerovu shemu

$$n = n \cdot 1 \qquad M(n) = n \cdot (n-1) + n \cdot \underbrace{M(1)}_{=\emptyset} = n \cdot (n-1).$$

- Ostaje, naravno, ključno pitanje: što se više isplati, faktorizirati n i kako, ili koristiti Hornerovu shemu, tj. kako treba faktorizirati n tako da dobijemo

$$\min M(n)$$

gdje $\min$ ide po svim faktorizacijama od n.

Prethodni rastav $n = n \cdot 1$ pokazuje da faktore jednake 1 na kraju možemo ignorirati i svesti na Hornerovu shemu.

Neka je $H(n) = n \cdot (n-1)$ funkcija koja opisuje broj množenja u Hornerovoj shemi za DFT_n.

- Pokazali smo da za bilo koji rastav $n = n_1 \cdot n_2$, $n_1, n_2 \geq 1$, vrijedi

$$M(n) = M(n_1 \cdot n_2) = n \cdot (n_1 - 1) + n_1 \cdot M(n_2).$$

Kad bismo za zadnji faktor n_2, za računanje DFT_{n_2} iskoristili Hornerovu shemu (što moramo, ako je n_2 prost),

$$M(n_2) = H(n_2) = n_2 \cdot (n_2 - 1)$$

dobili bismo

$$M(n) = n \cdot (n_1 - 1) + n_1 \cdot H(n_2) = n \cdot (n_1 - 1) + \underbrace{n_1 \cdot n_2}_{=n} (n_2 - 1)$$

$$= n \cdot [(n_1 - 1) + (n_2 - 1)].$$

Kada je to bolje od obične Hornerove sheme $H(n)$? Izračunajmo razliku $H(n) - M(n)$:

$$H(n) - M(n) = n \cdot \underbrace{(n-1)}_{=n_1 \cdot n_2} - n \cdot [(n_1 - 1) + (n_2 - 1)]$$

$$= n \cdot [n_1 n_2 - \cancel{1} - n_1 + \cancel{1} - n_2 + 1]$$

$$= n \cdot [n_1 \cdot n_2 - n_1 - n_2 + 1]$$

$$= n \cdot (n_1 - 1)(n_2 - 1)$$

Znamo da je $n, n_1, n_2 \geq 1$, pa je desna strana sigurno nenegativna tj. vrijedi

$$H(n) - M(n) = n \cdot (n_1 - 1)(n_2 - 1) \geq 0$$

ili

$$\underline{M(n) \leq H(n)}$$

za <u>svaki</u> $n \in \mathbb{N}$ i za <u>svaki</u> rastav $n = n_1 \cdot n_2$, $n_1, n_2 \geq 1$. ($n_1, n_2 \in \mathbb{N}$).

Odmah vidimo da se jednakost $M(n) = H(n)$ dostiže ako i samo ako je

$$n_1 = 1 \text{ ili } n_2 = 1.$$

Kad to primijenimo na naš rekurzivni "algoritam" (ili pristup) za računanje DFT_n dobivamo sljedeći zaključak.

Ako je $n \in \mathbb{N}$ složen broj, onda bilo kojom faktorizacijom

$$n = n_1 \cdot n_2, \quad n_1, n_2 > 1,$$

dobivamo rekurzivni algoritam za računanje DFT_n koji je brži (tj. ima manje kompleksnih aritmetičkih operacija) od Hornerove sheme za DFT_n.

Ovakvo rekurzivno ubrzanje nije moguće ako i samo ako je $n = 1$ ili $n = p$ prost broj.

- Ostaje još pitanje kako treba rastaviti (složeni) zadani broj n da dobijemo najmanji mogući broj množenja $M(n)$. Odgovor na to pitanje kaže kako treba organizirati nivoe rekurzije u rekurzivnom DFT_n algoritmu, tako da dobijemo najbrži mogući algoritam (tzv. Fast Discrete Fourier Transform, ili FFT_n).

Pretpostavimo da smo $n \in \mathbb{N}$ rastavili na faktore u obliku

$$n = n_1 \cdot n_2 \cdot \ldots \cdot n_q$$

gdje je $q \in \mathbb{N}$ i $n_1, \ldots, n_q \geq 1$. Lako se vidi da broj množenja u pripadnom rekurzivnom DFT_n algoritmu iznosi

$$M(n) = n \cdot [(n_1 - 1) + (n_2 - 1) + \ldots + (n_q - 1)]$$

i to bez obzira na "poredak" faktora, odnosno način realizacije rekurzije, sve dok za najdublji nivo koristimo Hornerovu shemu.

Jedan od načina organizacije rekurzije je redom

$$DFT_n \to \underbrace{DFT_{n_2 \cdot \ldots \cdot n_q}}_{n_1 \times} \to \underbrace{DFT_{n_3 \cdot \ldots \cdot n_q}}_{(n_1 \times) n_2 \times} \to \ldots \to \underbrace{DFT_{n_q}}_{(n_1 \times \ldots) \times n_{q-1} \times}$$

Treba naći najmanju međuvrijednost $\min M(n)$, po svim takvim rastavima $n = n_1 \cdot \ldots \cdot n_q$, za sve q. Označimo

$$M^*(n) = \min_{\substack{n = n_1 \cdot \ldots \cdot n_q \\ \text{i po } q}} M(n)$$

Vidimo da $M(n)$ uvijek ima faktor n, pa označimo

$$M(n) = n \cdot m(n), \quad m(n) = (n_1 - 1) + \dots + (n_g - 1)$$

za dani rastav $n = n_1 \cdot \dots \cdot n_g$. Treba naći

$$m^*(n) = \min m(n)$$

po svim rastavima od n na faktore, jer očito vrijedi

$$M^*(n) = n \cdot m^*(n).$$

- Znamo već da rekurzivni pristup nema prednosti pred Hornerovom shemom, ako (i samo ako) je $n = 1$ ili $n = p$ prost. Dakle, znamo:

$$m^*(n) = \begin{cases} \emptyset & , \text{ za } n = 1 \\ p - 1 & , \text{ za } n = p \text{ prost.} \end{cases}$$

- Već smo dokazali da za $n = n_1 \cdot n_2$, uz $n_1, n_2 > 1$, vrijedi

$$M(n) < H(n)$$

ili

$$m(n) = (n_1 - 1) + (n_2 - 1) < n_1 \cdot n_2 - 1 = n - 1.$$

To sugerira (čak diktira) svaki složeni faktor treba rastavljati dok god to možemo, a to znači na proste faktore.

Dakle, tvrdimo da se $m^*(n)$ postiže na rastavu od n na proste faktore, za svaki $n \geq 2$. (Naime, $n = 1$ po definiciji nije prost, a $m^*(1) = 0$ ionako znamo!)

Po istom principu, pretpostavimo da je neki faktor n_i složen, u rastavu $n = n_1 \cdot \dots \cdot n_g$, za neki $i \in \{1, \dots, g\}$ (Naravno, tada je i n složen). Ovom rastavu odgovara

$$m(n_1 \cdot \dots \cdot n_i \cdot \dots \cdot n_g) = (n_1 - 1) + \dots + (n_i - 1) + \dots + (n_g - 1).$$

Rastavimo li n_i na netrivijalne faktore $n_i = n_{i,1} \cdot n_{i,2}$, uz $n_{i,1}, n_{i,2} > 1$, onda je

$$m(n_1, \dots, n_{i,1}, n_{i,2}, \dots, n_g) = (n_1 - 1) + \dots + (n_{i,1} - 1) + (n_{i,2} - 1) + \dots + (n_g - 1)$$

No, zbog

$$(n_{i,1} - 1)(n_{i,2} - 1) > 0$$

$$\Rightarrow \quad (n_{i,1} - 1) + (n_{i,2} - 1) < n_{i,1} \cdot n_{i,2} = n_i$$

pa faktorizacijom dobivamo manju vrijednost funkcije m

$$m(n_1 \cdot \ldots \cdot n_{i,1} \cdot n_{i,2} \cdot \ldots \cdot n_g) < m(n_1 \cdot \ldots \cdot n_i \cdot \ldots \cdot n_g).$$

Odavde odmah slijedi da se $m^*(n)$ dostiže na rastavu od n koji ima samo <u>proste</u> faktore. U protivnom, rastavom bilo kojeg složenog faktora dobivamo manju vrijednost funkcije.

Znamo da svaki prirodni broj $n \geq 2$ možemo, i to jednoznačno, rastaviti u produkt prostih faktora

$$n = p_1^{\alpha_1} \cdot \ldots \cdot p_t^{\alpha_t}$$

gdje su $p_1 < \ldots < p_t$ prosti brojevi i $\alpha_1, \ldots, \alpha_t > 0$ prirodni eksponenti ($p_i^{\alpha_i} = 1$ za $\alpha_i = 0$, iznad ne igra ulogu). Također, znamo da poredak faktora za rekurzivni DFT_n uzet bitan i $m(n)$, pa onda i $m^*(n)$, ne ovise o poretku faktora (komutativnost zbroja).

Dalje, za $n \geq 2$ vrijedi

$$m^*(n) = \alpha_1(p_1 - 1) + \ldots + \alpha_t(p_t - 1).$$

(Iznad znamo $m^*(1) = 0$).

Drugim riječima, optimalni rekurzivni DFT_n, tj. FFT_n dobivamo kad n rastavimo na <u>proste</u> faktore i <u>tada</u> za broj množenja vrijedi

$$M^*(n) = n \cdot [\alpha_1(p_1 - 1) + \ldots + \alpha_t(p_t - 1)].$$

($g = \alpha_1 + \ldots + \alpha_g > 0$).

- Zaključujemo da brza varijanta diskretne Fourierove transformacije FFT_n "postoji" za svaki složeni prirodni broj n (u smislu da je FFT_n brži od Hornerove sheme).

- Na kraju, usporedimo ovaj rezultat s najpopularnijim "klasičnim" izborom za n, a to je kad je n potencija od 2

$$n = 2^m, \quad m \in \mathbb{N}_0.$$

($t = 1$, $p_1 = 2$, $\alpha_1 = m$).

Tada je $m^*(2^m) = m \cdot \underbrace{(2-1)}_{1} = m = \lg n$

i

$$M^*(n) = M^*(2^m) = n \cdot m = n \cdot \lg n$$

(Napomena: ovo odgovara sporijoj varijanti DFT_{2^m}, bez pomoćne varijable i odvrtanja. O dodatnim uštedama malo kasnije).

Naravno, $f(n) = n \cdot \lg n$, je kao funkcija, korektno definirana za $\forall n \in \mathbb{N}$, pa možemo uspoređivati $M^*(n)$ i $f(n) = n \cdot \lg n$ za sve n.

Upravo smo vidjeli

$$M^*(n) = n \cdot \lg n = f(n), \text{ za } n = 2^m, m \in \mathbb{N}_0.$$

Koji je odnos za ostale n, kad n nije potencija od 2. Treba usporediti $m^*(n)$ i $\lg n$, u rastavu n na proste faktore

$$n = p_1^{\alpha_1} \cdot \ldots \cdot p_t^{\alpha_t}.$$

Tada je

$$\lg n = \alpha_1 \cdot \lg p_1 + \ldots + \alpha_t \cdot \lg p_t$$

i

$$m^*(n) = \alpha_1 \cdot (p_1 - 1) + \ldots + \alpha_t (p_t - 1).$$

No, odmah vidimo da je

$$\lg n \leq n - 1$$

za svaki $n \in \mathbb{N}$, s tim da se jednakost dostiže za $n = 1$ i $n = 2$. Osim $p_1 = 2$, svi ostali prosti brojevi su veći od 2

$$t \geq 2 \Rightarrow p_t > 2 \Rightarrow \lg p_t < p_t - 1$$

pa iz $\alpha_t > 0 \Rightarrow$

$$m^*(n) < \lg n$$

čim n ima faktor $p_t > 2$.

U tom smislu, zaključujemo da "najbrži" mogući FFT_n dobivamo kad je n potencija od 2. Upravo zato se $n = 2^m$ koristi kad god je to moguće.

Nažalost, to ne ide uvijek. Naime $A(x)$ ili vektor a je lako dopuniti nulama do prve potencije od 2. Međutim, ta promjena n u 2^m mijenja i točke ω_n^k u kojima

se računa DFT, pa ne dobivamo iste mjedusobi! U nekim primjenama nas to ne smeta (na primjer, za brzo množenje polinoma), ali u drugima su i te tocke bitne. Kako tada treba postupiti - malo kasnije. Pokazat cemo da se, slicnim putem kao kod brzog množenja polinoma, putem konvolucije, može za bilo koji n dobiti DFT_n u $O(n \lg n)$ kompleksnih množenja.

Na kraju, odgovorimo na pitanje da li se u opcem rekurzivnom DFT_n algoritmu može napraviti slicna ušteda polovine množenja kao i u DFT_{2^m}.

Druga faza rekurzivnog algoritma (nakon rekurzivnih poziva) je računanje

$$y_k = y^{[\phi]}_{k \bmod n_2} + \sum_{\ell=1}^{n_1-1} \omega_n^{\ell \cdot k} \cdot y^{[\ell]}_{k \bmod n_2}, \quad k = 0, \dots, n-1.$$

Ušteda treba napraviti, ako ikde, koristeci neke pravilnosti i relacije za

$$\omega_n^{\ell \cdot k}$$

vezane uz ostatke modulo n_2. (ℓ je već $= j \bmod n_1$).
Napišimo stoga k u obliku

$$k = r + s \cdot n_2, \quad \begin{cases} r = 0, \dots, n_2 - 1 \\ s = 0, \dots, n_1 - 1, \end{cases}$$

pa je $k \bmod n_2 = r$. Onda je

$$\omega_n^{\ell \cdot k} = \omega_n^{\ell \cdot (r + s \cdot n_2)} = \omega_n^{\ell \cdot r} \cdot \underbrace{\omega_n^{\ell \cdot s \cdot n_2}}_{n_1 \cdot n_2} = \omega_n^{\ell \cdot r} \cdot \omega_{n_1}^{\ell \cdot s}$$

pa je:

$$y_{r+s \cdot n_2} = y^{[\phi]}_r + \sum_{\ell=1}^{n_1-1} \underbrace{\omega_{n_1}^{\ell \cdot s}}_{\text{ovisi samo o } n_1} \cdot \omega_n^{\ell \cdot r} \cdot y^{[\ell]}_r, \quad \begin{cases} r = 0, \dots, n_2 - 1, \\ s = 0, \dots, n_1 - 1. \end{cases}$$

(potencijalna ušteda)

Ove izraze - produkte računamo u dvije faze. (Na zbrajanju, očito, nećemo moći napraviti uštedu — sve sumande će trebati zbrojiti!)

(A): Prvo računamo sve one produkte koji ne ovise o s, već samo o r i ℓ:

$$z_r^{[\ell]} = \omega_n^{\ell \cdot r} \cdot y_r^{[\ell]}, \quad \begin{cases} \ell = 1, -, n_1 - 1 \\ r = 0, -, n_2 - 1 . \end{cases}$$

Tih produkata ima $(n_1 - 1) \cdot n_2$.

Onog za $\ell = 0$ ionako ne računamo – pripadni član je već ispred sume. Ako izvršimo sva ova množenja, broj množenja u fazi A je

$$(n_1 - 1) \cdot n_2 \ M.$$

Na produktima za $r = 0$ je $z_0^{[\ell]} = y_0^{[\ell]}$, pa bismo mogli uštedjeti po jedno množenje i ukupni broj množenja bi bio

$$(n_1 - 1)(n_2 - 1) \ M.$$

To bi odgovaralo (v. malo niže) uštedi svih množenja u računanju y_0, tj. za $k = 0$, a tu (očitu) uštedu ni ranije nismo brojali (smanjili $M(n)$), pa nećemo ni sada inzistirati na njoj.

(B): Zatim izračunamo produkte

$$z_{r,s}^{[\ell]} = \omega_{n_1}^{\ell \cdot s} \cdot z_r^{[\ell]}, \quad \begin{cases} \ell, r \text{ kao gore u (A)} \\ s = 1, -, n_1 - 1, \end{cases}$$

s tim da konstruiramo da je $\omega_{n_1}^{\ell \cdot s} = 1$ za $s = 0$, i definiramo: $z_{r,\emptyset}^{[\ell]} = z_r^{[\ell]}$.

Produkata $z_{r,s}^{[\ell]}$, $s \neq 0$ ima $(n_1 - 1)^2 \cdot n_2$.

(C): Na kraju, za dovršenje druge faze, treba izračunati još i sume

$$y_{r+s \cdot n_2} = y_r^{[\emptyset]} + \sum_{\ell=1}^{n_1 - 1} z_{r,s}^{[\ell]}, \quad \begin{cases} r = 0, -, n_2 - 1 \\ s = 0, -, n_1 - 1 . \end{cases}$$

(tu više nema množenja).

Ako produkte u fazi B izračunamo tako da zaista množimo sve navedene mnjednosti (faktore), onda imamo

$$(n_1-1)^2\, n_2\, M$$

množenja u fazi B. Ovdje ne pomaže ni $r=\emptyset$, jer je $z_0^{[e]} = y_0^{[e]}$, a to može biti bilo što, pa pripadnih množenja ima.

U tom slučaju, ukupno bismo za dvije faze A i B zajedno imali sljedeći broj kompleksnih množenja:

$$\underbrace{(n_1-1)\cdot n_2}_{\text{faza A}} + \underbrace{(n_1-1)^2 n_2}_{\text{faza B}} = (n_1-1)\cdot n_2 \cdot \underbrace{[1+(n_1-1)]}_{n_1}$$

$$= n_1 \cdot n_2 \cdot (n_1-1) = n\cdot(n_1-1),$$

a to je isto kao i ranije u Hornerovoj shemi u fazi 2.

Čak i kad bismo za fazu A ureli "pedantni" broj od $(n_1-1)(n_2-1)$ množenja, još uvijek dobivamo:

$$(n_1-1)(n_2-1) + (n_1-1)^2 n_2 = (n_1-1)\cdot[\, n_2 - 1 + (n_1-1)\cdot n_2\,]$$

$$= (n_1-1)\cdot[\,\cancel{n_2} - 1 + n_1\cdot n_2 - \cancel{n_2}\,]$$

$$= (n_1\cdot n_2 - 1)\cdot(n_1-1)$$

$$= (n-1)\cdot(n_1-1).$$

Ovo točno odgovara ranijoj potencijalnoj uštedi od n_1-1 množenja u računanju y_0, za $k=0$ u Hornerovoj shemi. Dakle, ne dobivamo ništa bitno novo i bolje.

Gdje je onda ušteda? Još uvijek nije vidljiva!

Ideja je pametno kombinirati faze B i C, tako da ne moramo izračunati baš sve produkte u fazi B, već pokušavamo neke izraziti preko drugih i to umetnuti u C.

- Pogledajmo prvo situaciju za $n_1 = 2$, neovisno o n_2, tj. nije bitno da n bude potencija od 2, već samo $n = 2 \cdot n_2$ (prvi prosti faktor je 2).

Tada je očito $$\omega_{n_1} = \omega_2 = -1$$

pa je blesavo množiti s potencijama od -1, kad to možemo realizirati oduzimanjem u fazi C, ako je eksponent baš negativan.

U fazi B je tada $\ell = 1$ (drugih ℓ-ova nema, jer je $n_1 - 1 = 1$, a član za $\ell = 0$ je već na početku sume u fazi C). Isto vrijedi za s-tog "produktnih" članova – imamo samo

$$z_{r,1}^{[1]} = \omega_2^{1\cdot 1}\, z_r^{[1]} = -z_r^{[1]}$$

dok je $$z_{r,0}^{[1]} = z_r^{[1]}$$

po definiciji.

Dakle, množenja u fazi B uopće nisu potrebna ako fazu C realiziramo u paru $s = 0, 1$, kao

$$\left.\begin{aligned} (s=\emptyset):\quad & y_r = y_r^{[\emptyset]} + z_{r,0}^{[1]} = y_r^{[\emptyset]} + z_r^{[1]} \\ (s=1):\quad & y_{r+n_2} = y_r^{[\emptyset]} + z_{r,1}^{[1]} = y_r^{[\emptyset]} - z_r^{[1]} \end{aligned}\right\} \quad r = 0, \dots, n_2 - 1.$$

Imamo samo $(n_1 - 1)\cdot n_2 = \frac{n}{2}$ množenja u fazi A i štedimo $(n_1-1)^2 n_2 = \frac{n}{2}$ množenja u fazi B.

Štedimo <u>polovinu</u> množenja pri prijelazu iz $DFT_{n/2}$ na DFT_n, čim je $n_1 = 2$. Dakle

$$M(2\cdot n_2) = \frac{n}{2} + 2\cdot M(n_2)$$

a ne $n + 2M(n_2)$. Naravno, to vrijedi za svaki faktor $p_1 = 2$ u rastavu od n na proste faktore, tj. za

$$n = 2^{\alpha_1} \cdot p_i^{\alpha_i} \cdot p_t^{\alpha_t}$$

je broj kompleksnih množenja jednak

$$\begin{aligned} M'(n) &= \frac{1}{2} n\cdot\alpha_1 + n\cdot\left[\alpha_i(p_i - 1) + \dots + \alpha_t(p_t - 1)\right] \\ &= \frac{1}{2} n\cdot\alpha_1 + M\left(\frac{n}{2^{\alpha_1}}\right). \end{aligned}$$

Broj zbrajanja ostaje isti kao i prije $M(n) = n\cdot\alpha_1 + M\left(\frac{n}{2^{\alpha_1}}\right)$.

Ako je n potencija od 2, tj. $n = 2^m$, $m \in \mathbb{N}_0$, onda je odmah vidljivo da je broj operacija u staroj brzoj algoritmu za DFT_{2^m} jednak

$\frac{1}{2} n \cdot \lg n$ kompleksnih množenja

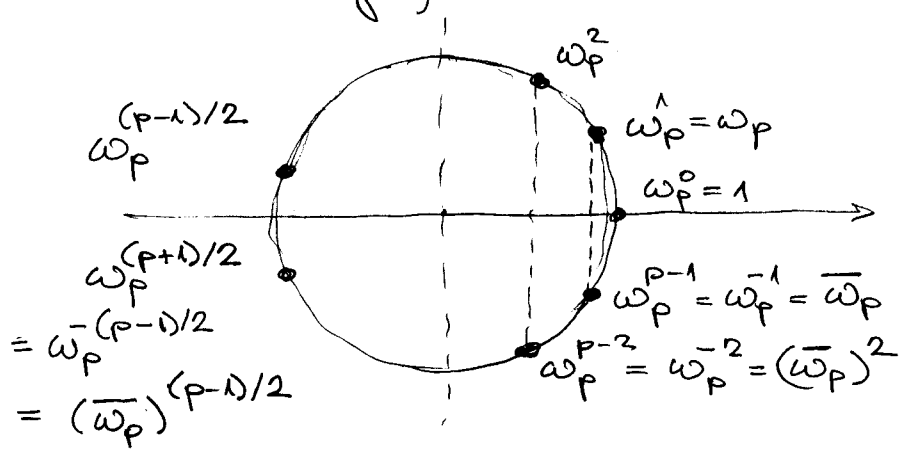
$n \cdot \lg n$ kompleksnih zbrajanja.

– Ostaje da li nešto slično možemo napraviti i kad je $n_1 > 2$.

Prvo uočimo da je, zapravo, dovoljno gledati slučaj kad je $n_1 = p > 2$ neparan prost broj, tj. FFT_n (dalje, najbrži oblik algoritma za DFT_n), uz $n = p \cdot n_2$.

[Ovo nećemo bitno iskoristiti u nastavku, sve ide za bilo koji $n_1 > 2$.]

Tada su potencije ω_p^q simetrično raspoređene po jediničnoj kružnici, obzirom na realnu os



[Ovo je izgled za neparne p. Isto imamo za neparne n_1, a za parne n_1 je $\omega_{n_1}^{n_1/2} = (-1)$ pa taj nema svoj par, ali njega spojimo s $\omega_{n_1}^0 = 1$, kao za $n_1 = 2$].

Obzirom na to da suma ide po l, treba iskoristiti simetriju po s. Vidimo da treba gledati s i $p_1 - s$, s tim da s ide od 1 do $(p_1 - 1)/2$, za $n_1 = p$ neparan (prost).

Nažalost, tada je $\omega_p^q \neq \pm 1$ za $q = 1, \dots, p-1$ pa simetriju ne možemo iskoristiti na nivou kompleksne aritmetike (nema ni simetrije obzirom na imaginarnu os !! – dokažite).

Pogledajmo ovdje realizaciju kompleksnih aritmetičkih operacija preko realnih aritmetičkih operacija.

Općenito je za jedno kompleksno množenje potrebno 4 realna množenja i 2 realna zbrajanja

$$1M = 4M_R + 2A_R$$

iz $(e+fi) = (a+bi)\cdot(c+di) =$
$= (ac-bd) + i(ad+bc)$

tj. $e = ac - bd$ $\quad 2M_R + 1A_R$
$f = ad + bc$ $\quad -\,''\,-$.

U fazi B gledamo par proodukata

$$z^{[e]}_{r,s} = \omega_p^{e\cdot s}\cdot z^{[e]}_r$$

$$z^{[e]}_{r,p-s} = \omega_p^{e\cdot(p-s)}\cdot z^{[e]}_r$$
$$= \underbrace{\omega_p^{e\cdot p}}_{=1}\cdot\underbrace{\omega_p^{-e\cdot s}}_{(\overline{\omega}_p)^{e\cdot s}}\cdot z^{[e]}_r$$
$$= (\overline{\omega}_p)^{e\cdot s}\cdot z^{[e]}_r = \overline{(\omega_p^{e\cdot s})}\cdot z^{[e]}_r$$

za $s = 1, -, (p-1)/2$. Direktni račun bez ušteda treba $2M$ za ove produkte (i još $2A$ za ponpadna zbrajanja u fazi C), tj.

$$2M = 8M_R + 4A_R$$
$$(+2A = \quad 4A_R)$$

No, faktori $\omega_p^{e\cdot s}$ i $\overline{(\omega_p^{e\cdot s})}$ imaju iste realne i suprotne imaginarne dijelove. Usporedbom relacija

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

vidimo odmah da trebamo samo 4, a ne 8, realnih množenja (i 4 realna zbrajanja ili oduzimanja).

Dalje, za par $z^{[e]}_{r,s}$, $z^{[e]}_{r,p-s}$,

$e = \textcircled{1}, -, n_1 - 1 \quad (p-1)$
$r = 0, -, n_2 - 1$
$s = \textcircled{1}, -, \frac{n_1-1}{2} \quad (\frac{p-1}{2})$

treba $4M_R + 4A_R$ po paru,
što daje uštedu od polovine svih realnih množenja u fazi B (a broj realnih zbrajanja ostaje isti).

Ukupno gledajući, uismo uštedili baš polovinu svih realnih umnožavanja, jer u fazi A nema uštede. Nb, faza A zna za red veličine manje produkata.

$$A: \quad (p-1)\cdot\frac{n}{p}\,M \iff 4(p-1)\cdot\frac{n}{p}\,M_R$$

$$B: \quad \text{ušteda} \iff 2(p-1)^2\cdot\frac{n}{p}\,M_R$$

↑ umjesto 4

pa je ušteda bliže $\frac{1}{2}$ ukupnog broja realnih umnožavanja.

U prvoj varijanti imali smo $(1M \to 4M_R)$

$$4n\cdot(p-1)\;M_R$$

a sada imamo

$$4(p-1)\cdot\frac{n}{p} + 2(p-1)^2\cdot\frac{n}{p} = 2n\cdot(p-1)\left[\frac{2}{p} + \frac{p-1}{p}\right]$$

$$\frac{p+1}{p}$$

$$= \underbrace{\left(1+\frac{1}{p}\right)}_{\substack{\text{ovo je} \\ \text{"malo"} \\ \text{veće od 1.}}} \cdot \underbrace{2n(p-1)}_{\substack{\text{ovo bi bila polovina} \\ \text{ranijeg broja realnih umnožavanja.}}} M_R .$$

- Pokažite da isti argument vrijedi za bilo koji neparni n_1 i samo treba promijeniti $p \mapsto n_1$ u zadnjem broju umnožavanja.
- Što se dobije za n_1 paran? [Isto što i za $n_1 = 2$, tj. točno polovina realnih umnožavanja].

- Znamo da kompleksno množenje možemo realizirati i ovako: $(e+fi) = (a+bi)(c+di)$

$$\left.\begin{aligned} t_1 &= ac \\ t_2 &= bd \\ t_3 &= (a+b)(c+d) \\ e &= t_1 - t_2 \\ f &= t_3 - t_1 - t_2 \end{aligned}\right\}$$

što daje $1M = 3M_R + 5A_R$ (ušteda je $1M_R$ na račun $3A_R$).

Što se tada događa za $n_1 = 2$, $n_1 = p$, n_1 neparan, n_1 paran? ($p > 2$)