Fast Fourier Transform

Παναγιώτης Πατσιλινάκος

ΕΜΕ

19 Οκτωβρίου 2017

Πολλαπλασιασμός Πολυωνύμων

- Στόχος μας είναι ο πολλαπλασιασμός πολυωνύμων.
- Για παράδειγμα, για δυο πολυώνυμα βαθμού d, $A(x) = a_0 + a_1x + ... + a_dx^d$ και $B(x) = b_0 + b_1x + ... + b_dx^d$, ψάχνουμε το $C(x) = A(x) \cdot B(x) = c_0 + c_1x + ... + c_{2d}x^{2d}$
- Ουσιαστικά ψάχνουμε τους συντελεστές, όπου

$$c_k = a_0b_k + a_1b_{k-1} + ... + a_kb_0 = \sum_{i=0}^{k} a_ib_{k-i}$$

Όπου για $i > d$ θεωρούμε τα a_i και b_i μηδέν και $0 \leq k \leq 2d$.

- Από τον παραπάνω τύπο, θα είχαμε $O(k)$ βήματα για τον υπολογισμό ενός συντελεστή και άρα $O(d^2)$ για τον υπολογισμό των $2d + 1$ συντελεστών.

Γιατί;

- Για πολλούς λόγους.
- Πολλαπλασιασμός ακεραίων.
- Ουσιαστικά, αν θέσουμε όπου x τη βάση 2 και θεωρήσουμε δυαδικούς συντελεστές, έχουμε το γινόμενο δυαδικών αριθμών.
- (Στην ανάλυση, θα θεωρούμε ότι το γινόμενο δυο συντελεστών γίνεται σε ένα βήμα. Αυτό δεν επηρεάζει την μελέτη του γινομένου δυαδικών.)
- Πρακτικά μιλάμε για τη συνέλιξη δυο διανυσμάτων $a * b$ για $a, b \in \mathbb{R}^n$, όπου:

$$a * b = \sum_{\substack{(i,j):i+j=k \\ i<m, j<n}} a_i b_j$$

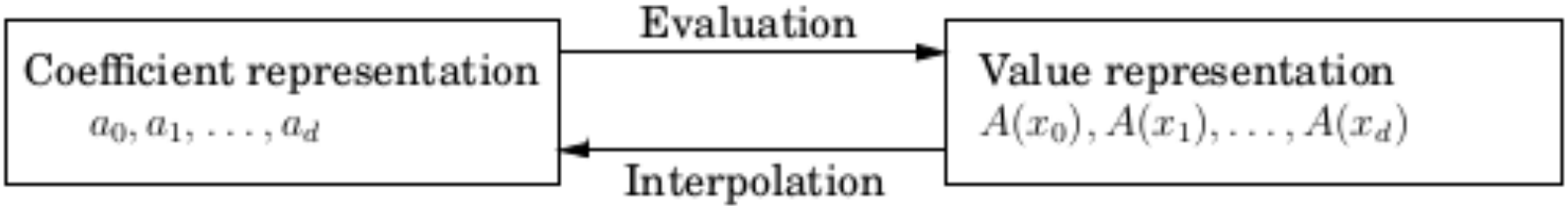
Δηλαδή : $a * b = (a_0 b_0, a_0 b_1 + a_1 b_0, a_0 b_2 + a_1 b_1 + a_2 b_0, \ldots, a_{n-2} b_{n-1} + a_{n-1} b_{n-2}, a_{n-1} b_{n-1})$

Συμβολισμός

- Κάθε πολυώνυμο βαθμού d χαρακτηρίζεται μοναδικά από τις τιμές του σε οποιαδήποτε $d+1$ διακριτά σημεία.
- Άρα, για οποιαδήποτε διακριτά σημεία $x_0, ..., x_d$, μπορούμε να ορίσουμε ισοδύναμα ένα πολυώνυμο βαθμού d, $A(x) = a_0 + ... + a_d x^d$:
  - Από τους συντελεστές του $a_0, ..., a_d$
  - Τις τιμές του $A(x_0), ..., A(x_d)$ (αναπαράσταση τιμών)
- Αλλά, ένα πολυώνυμο $C(x)$ που είναι γινόμενο δυο πολυωνύμων βαθμού d, έχει βαθμό $2d$ και χαρακτηρίζεται πλήρως από την τιμή του σε $2d+1$ σημεία.
- Όμως η τιμή του σε ένα σημείο z μπορεί να υπολογιστεί σε 'ένα' βήμα, $A[z] \cdot B[z]$, αν έχουμε τις τιμές $A[z]$ και $B[z]$.
- Άρα ο πολλαπλασιασμός πολυωνύμων γίνεται σε γραμμικό χρόνο όταν γίνεται με αναπαράσταση τιμών.

Evaluation - Interpolation

- Το πρόβλημα είναι ότι θέλουμε σαν είσοδο και έξοδο να έχουμε πολυώνυμα σε αναπαράσταση με συντελεστές.
- Άρα, θα πρέπει να κάνουμε δυο μετατροπές:



- Όμως πόσο γρήγορα μπορούμε να κάνουμε αυτές τις μετατροπές;
- Αν αποτίμηση ενός πολυωνύμου βαθμού $d \leq n$ γενικώς απαιτεί $O(n)$ βήματα, τότε, για n σημεία, θα είχαμε πάλι χρόνο $O(n^2)$.
- Ο ταχύς μετασχηματισμός Fourier (FFT) κάνει τον παραπάνω υπολογισμό σε χρόνο $O(nlogn)$

Διαίρει και Βασίλευε Ι

- Θα διαλέξουμε έξυπνα τα σημεία που θα αποτιμήσουμε.
- Για πολυώνυμο βαθμού $\leq n-1$, διαλέγουμε ζευγάρια αριθμών με τους αντίθετους τους:

$$\pm x_0, \pm x_1, \ldots, \pm x_{n/2-1}$$

- Έτσι, για κάθε ζεύγος $A(x_i)$, $A(-x_i)$ θα κερδίσουμε από τους υπολογισμούς, χρησιμοποιώντας το γεγονός ότι οι ζυγές δυνάμεις του x_i είναι ίδιες με αυτές του $-x_i$.
- Για να το πετύχουμε αυτό, θα χωρίσουμε το πολυώνυμο $A(x)$ σε ζυγές και μονές δυνάμεις. Πχ:
  $3+4x+6x^2+2x^3+x^4+10x^5 = (3+6x^2+x^4)+x(4+2x^4+10x^4)$

Διαίρει και Βασίλευε II

- Έτσι χωρίσαμε το αρχικό πολυώνυμο σε πολυώνυμα του x^2.:

$$A(x) = A_e(x^2) + xA_o(x^2)$$

Όπου το A_e έχει τους συντελεστές των ζυγών δυνάμεων και το A_o των μονών του A αντίστοιχα.

- Τα A_e και A_o θα έχουν βαθμό $\leq n/2 - 1$
- Έτσι, οι υπολογισμοί που χρειάζονται για να υπολογιστεί το $A(x_i)$, μπορούν να επαναχρησιμοποιηθούν για τον υπολογισμό του $A(-x_i)$ καθώς:

$$A(x_i) = A_e(x_i^2) + x_i A(x_i^2)$$
$$A(-x_i) = A_e(x_i^2) - x_i A(x_i^2)$$

Διαίρει και Βασίλευε III

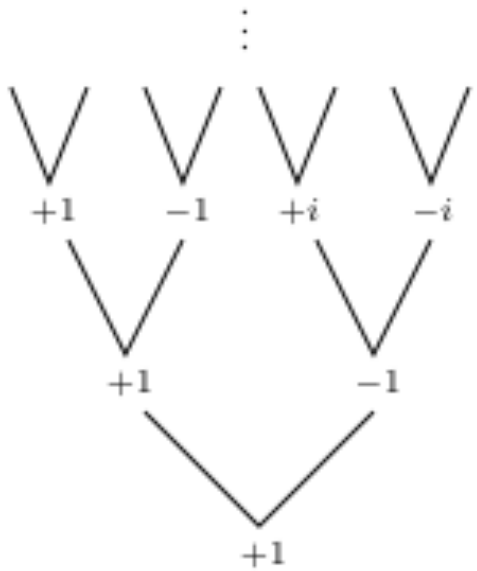
- Έτσι, ο υπολογισμός του $A(x)$ σε n ζεύγη σημείων $\pm x_0, \ldots \pm x_{n/2-1}$, ανάγεται στον υπολογισμό των $A_e(x)$, $A_o(x)$ στα $n/2$ σημεία $x_0^2, \ldots, x_{n/2-1}^2$
- Με αυτόν τον τρόπο, το αρχικό πρόβλημα μεγέθους n χωρίζεται σε δυο υποπροβλήματα ίδιου τύπου, μεγέθους $n/2$ που συνδυάζονται με αριθμητικές πράξεις γραμμικού χρόνου.
- Συνεπώς προκύπτει η αναδρομική σχέση:

$$T(n) = 2T(n/2) + O(n)$$

- Που είναι ίσο με $O(nlogn)$.

Επιλογή σημείων Ι

- Τα σημεία που έχουμε επιλέξει μέχρι στιγμής, λειτουργούν μόνο στο πρώτο επίπεδο της αναδρομής.
- Για να συνεχίσουμε στο επόμενο επίπεδο, θα θέλαμε τα $x_0^2, ..., x_{n/2-1}^2$ να είναι επίσης ζεύγη αριθμών με τους αντίθετους τους.
- Λύση: Μιγαδικοί αριθμοί.
- Στο τελευταίο επίπεδο της αναδρομής, μας μένει ένα μοναδικό σημείο. Αν επιλέξουμε αυτό να είναι ο αριθμός 1:



Επιλογή σημείων ΙΙ

- Τελικά ψάχνουμε τις n μιγαδικές ρίζες του $z^n = 1$.
- Οι ρίζες είναι οι $1, \omega, \omega^2, ..., \omega^{n-1}$ όπου $\omega = e^{2ni/n}$, με τις ιδιότητες:
  1. Οι n-οστές ρίζες είναι ζεύγη θετικών-αρνητικών αριθμών, $(\omega^{n/2+j} = -\omega^j)$
  2. 'Τετραγωνίζοντάς' αυτές, έχουμε τις $n/2$ ρίζες του $z^{n/2} = 1$
- Έτσι, για n δύναμη του 2, σε κάθε επόμενο βήμα της αναδρομής έχουμε τις $(n/2^k)$ ρίζες του $z^{n/2^k} = 1$, για $k = 0, 1, 2, ...$, κάθε σετ εκ' των οποίων είναι ζεύγη θετικών-αρνητικών αριθμών. Έτσι, ο αλγόριθμος λειτουργεί.

Αντιστροφή I

- Όπως είπαμε στην αρχή, θέλουμε έναν αλγόριθμο που να δέχεται πολυώνυμα με περιγραφή συντελεστών και να επιστρέφει το γινόμενό τους επίσης σε μορφή συντελεστών.
- Μέχρι στιγμής είδαμε πως να μετατρέψουμε την είσοδο σε αναπαράσταση τιμών και γνωρίζουμε ότι σε αυτήν την αναπαράσταση βρίσκουμε το γινόμενο σε γραμμικό χρόνο.
- Τώρα πρέπει να μετατρέψουμε το γινόμενο από αναπαράσταση τιμών σε περιγραφή τελεστών.

Αντιστροφή II

- Ο FFT, μετακινεί (περιστρέφει) τους συντελεστές σε τιμές, με είσοδο τις n ρίζες του $z^n = 1$, $(1, \omega, \omega^2, ..., \omega^{n-1})$:

$$\langle \text{τιμές} \rangle = FFT(\langle \text{συντελεστές} \rangle, \omega)$$

- Ουσιαστικά, έχουμε το γραμμικό μετασχηματισμό:

$$\begin{bmatrix} A(x_0) \\ A(x_1) \\ . \\ . \\ A(x_{n-1}) \end{bmatrix} = \begin{bmatrix} 1 & x_0 & x_0^2 & ... & x_0^{n-1} \\ 1 & x_1 & x_1^2 & ... & x_1^{n-1} \\ & & . & & \\ & & . & & \\ 1 & x_{n-1} & x_{n-1}^2 & ... & x_{n-1}^{n-1} \end{bmatrix} \begin{bmatrix} a_0 \\ a_2 \\ . \\ . \\ a_{n-1} \end{bmatrix}$$

- Αλλά ο μεσαίος πίνακας $(M_n(x))$ είναι πίνακας Vandermonde, συνεπώς αντιστρέψιμος (τα πολυώνυμα χαρακτηρίζονται μοναδικά με την αναπαράσταση τιμών).
- Στην ειδική περίπτωση του $M_n(\omega)$ ισχύει $M_n(\omega)^{-1} = \frac{1}{n} M_n(\omega^{-1})$

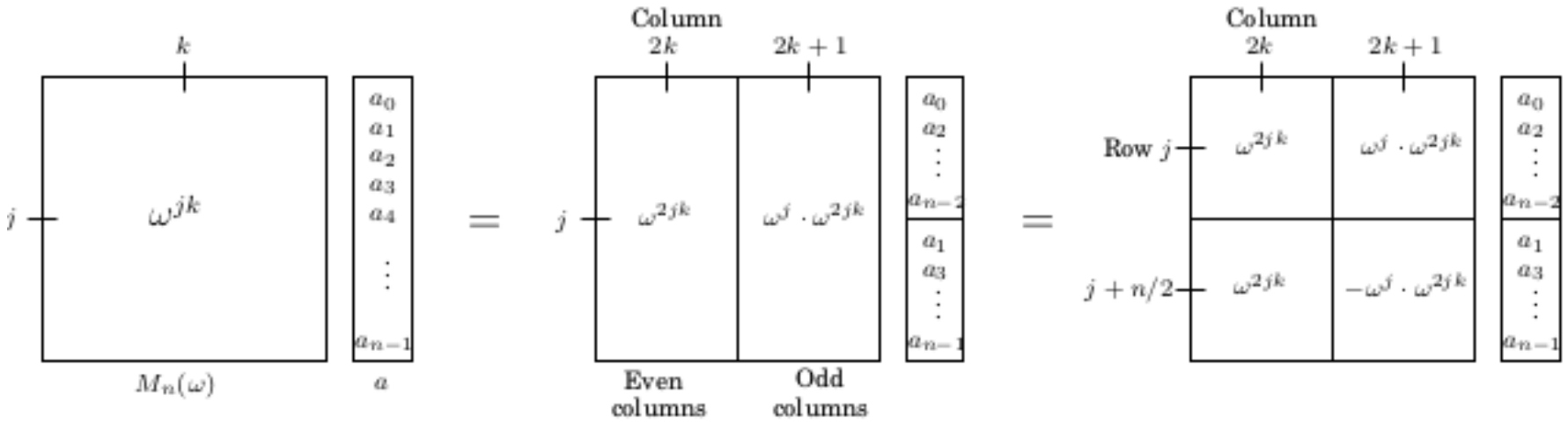
Αντιστροφή III

- Έτσι, για το interpolation έχουμε:

$$\langle \text{συντελεστές} \rangle = \frac{1}{n} FFT(\langle \text{τιμές} \rangle, \omega^{-1})$$

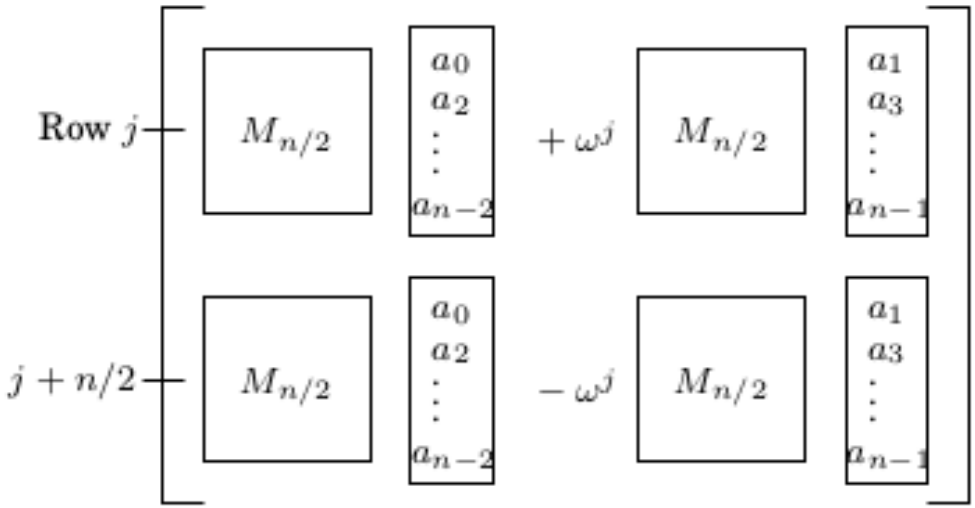
Περιγραφή του Αλγορίθμου I

- Όπως είδαμε, ο αλγόριθμος υπολογίζει ουσιαστικά το γινόμενο του πίνακα $M_n(\omega)$ με το διάνυσμα των συντελεστών $a = (a_0, ..., a_{n-1})$, ως είσοδο, (όπου ω ο αριθμός του οπίου οι δυνάμεις, μέχρι $n-1$ είναι οι ρίζες του $z^n = 1$.
- Αν αναδιατάξουμε τις στήλες του M για να ομαδοποιήσουμε τις ζυγές και τις μονές, και απλοποιήσουμε το κάτω μισό του πίνακα, χρησιμοποιώντας $\omega^{n/2} = -1$ και $\omega^n = 1$:



Περιγραφή του Αλγορίθμου II

- Βλέπουμε το επόμενο στάδιο της αναδρομής:



Όπου κάθε υποπίνακας $M_{n/2}$ είναι ο $M_{n/2}(\omega^2)$

Περιγραφή του Αλγορίθμου III

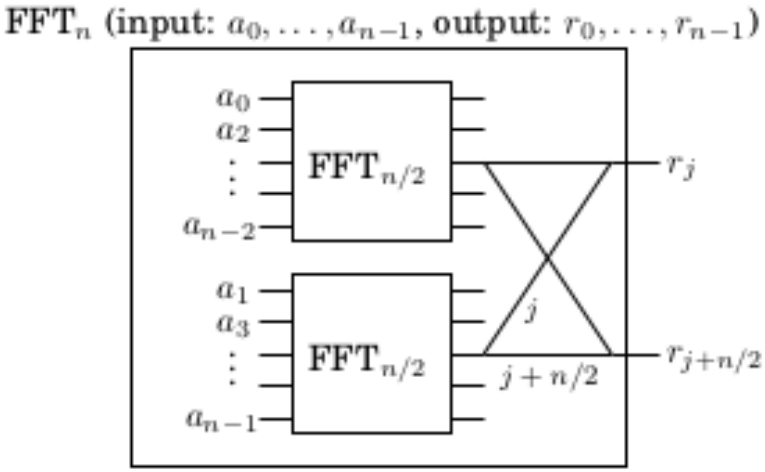
Algorithm 1 FFT(a, ω)

0: **Είσοδος**: Διάνυσμα $a = (a_0, ..., a_{n-1})$ για n δύναμη του 2, Η βάση ω των ριζών του $z^n = 1$
0: **Έξοδος**: Το γινόμενο $M_n(\omega)(a)$

1: **if** $\omega = 1$
    return a
2: $(s_0, s_1, ..., s_{n/2-1}) = FFT((a_0, a_2, ..., a_{n-2}), \omega^2)$
3: $(s'_0, s'_1, ..., s'_{n/2-1}) = FFT((a_1, a_3, ..., a_{n-1}), \omega^2)$
4: **for** $j = 0$ to $n/2 - 1$ **do**
5: $\quad r_j = s_j + \omega^j s'_j$
6: $\quad r_{j+n/2} = s_j - \omega^j s'_j$
7: **end for**
8: **return** $(r_0, r_1, ..., r_{n-1})$

Παράδειγμα Ι

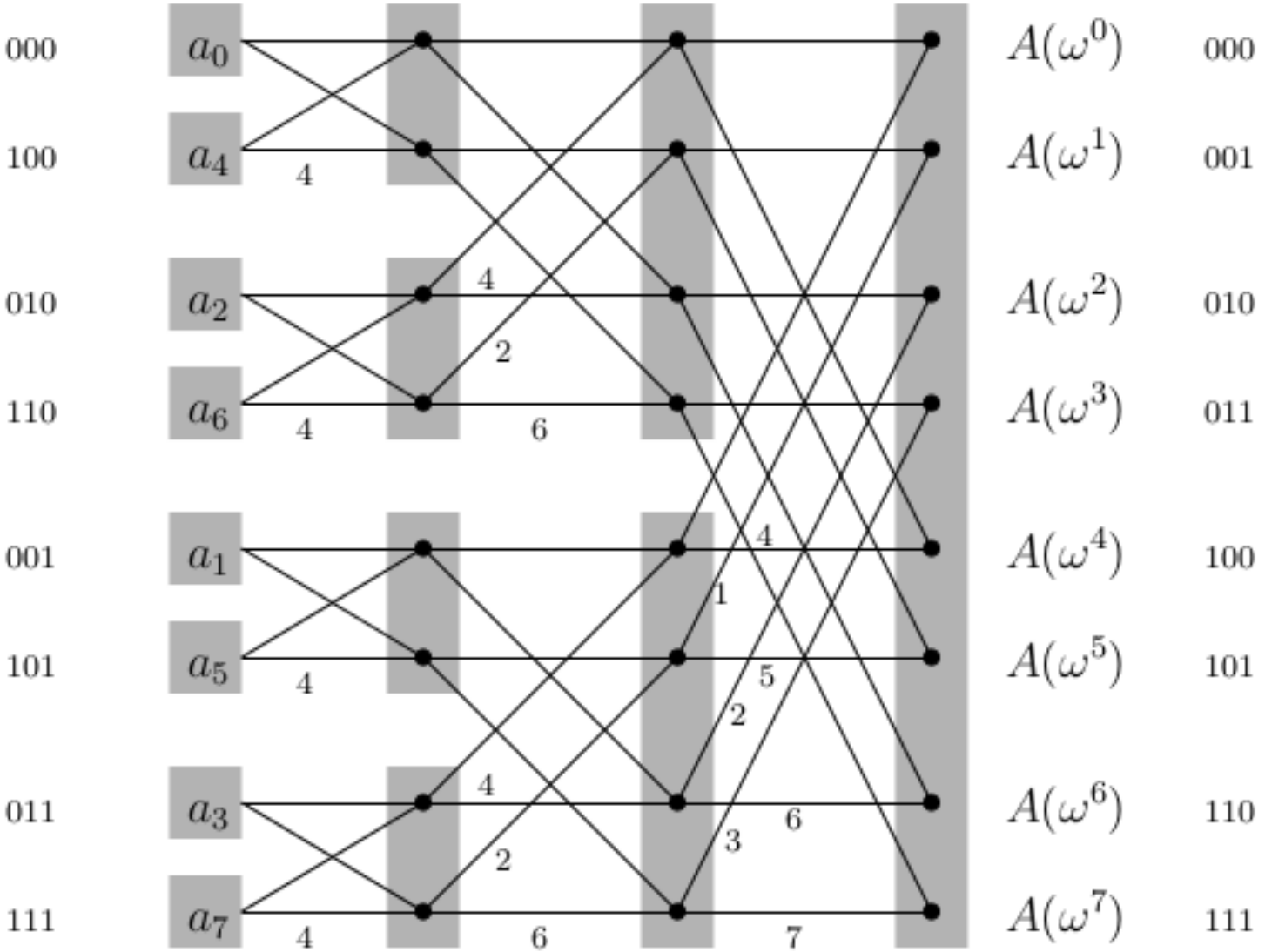
- Σαν κύκλωμα:



- Το εξωτερικό κουτί, λύνει το πρόβλημα για μέγεθος n, και τα εσωτερικά, για υποπροβλήματα μεγέθους $n/2$.
- Στην παραπάνω πεταλούδα, το βάρος j στην ακμή σημαίνει ότι πολλαπλασιάζουμε την πληροφορία που 'μεταφέρει' με ω^j, ενώ η τομή δυο ακμών από τα αριστερά σημαίνει πρόσθεση των περιεχομένων τους.

Παράδειγμα ΙΙ

- Για $n = 8$:



Βιβλιογραφία


S.Dasgupta,C.H.Papadimitriou, and U.V.Vazirani
Algorithms
McGraw-Hill, 2008


J. Kleinberg and E. Tardos
Algorithm Design
Addison-Wesley, 2006