Αλγόριθμοι και Πολυπλοκότητα

7ο εξάμηνο

Σ.Η.Μ.Μ.Υ. & Σ.Ε.Μ.Φ.Ε.

http://www.corelab.ece.ntua.gr/courses/

7η εβδομάδα: Οπισθοδρόμηση (Backtracking), Δέντρα Παιχνιδιών, Δίκτυα Ταξινόμησης

Διδάσκοντες:

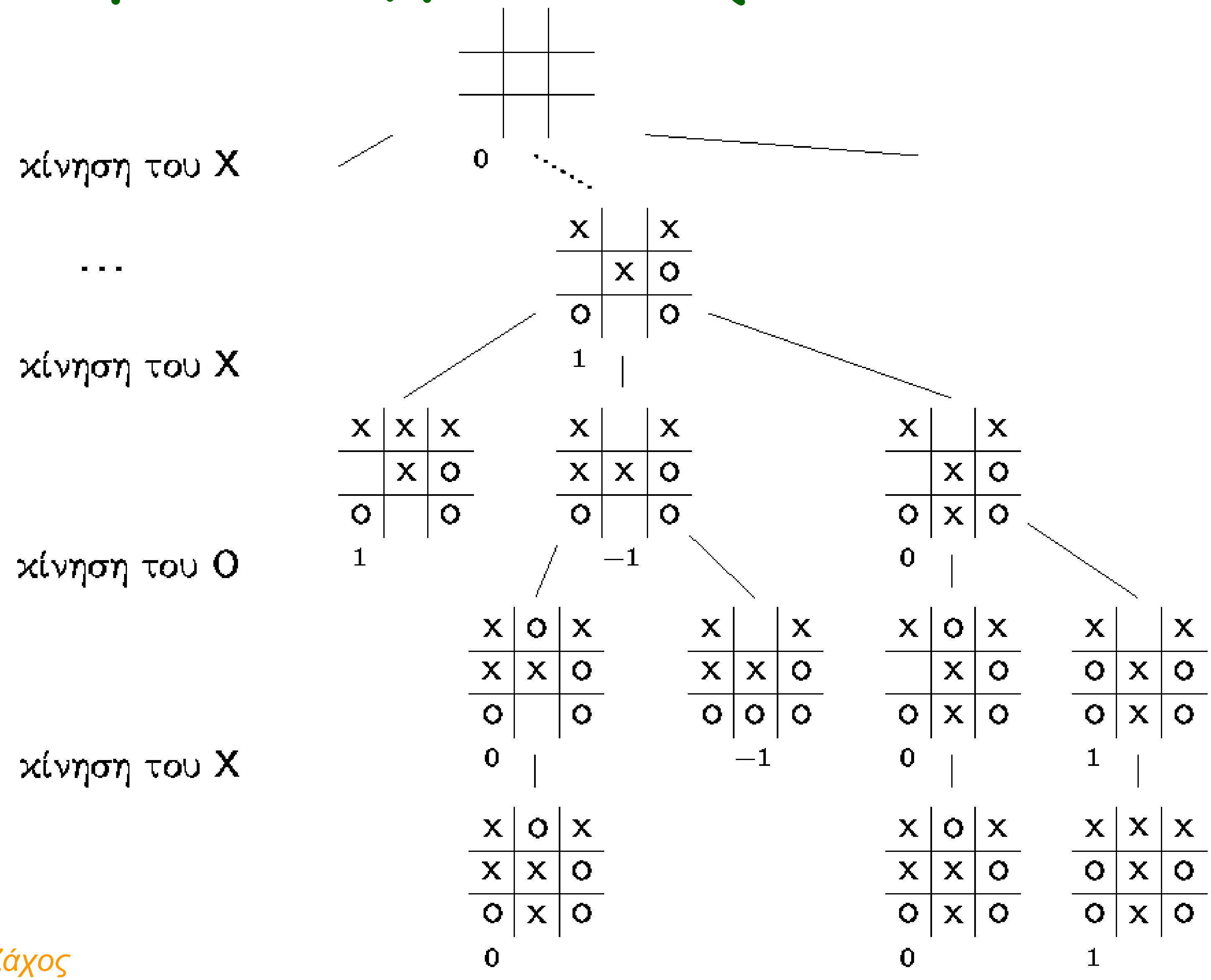
Στάθης Ζάχος - Άρης Παγουρτζής

Οπισθοδρόμηση (Backtracking)

- Συστηματική εξαντλητική αναζήτηση (exhaustive search).

- Τεχνικές μείωσης δέντρου επιλογών:
  A-B prunning, branch and bound.

Δέντρα Παιχνιδιών (Game Trees)

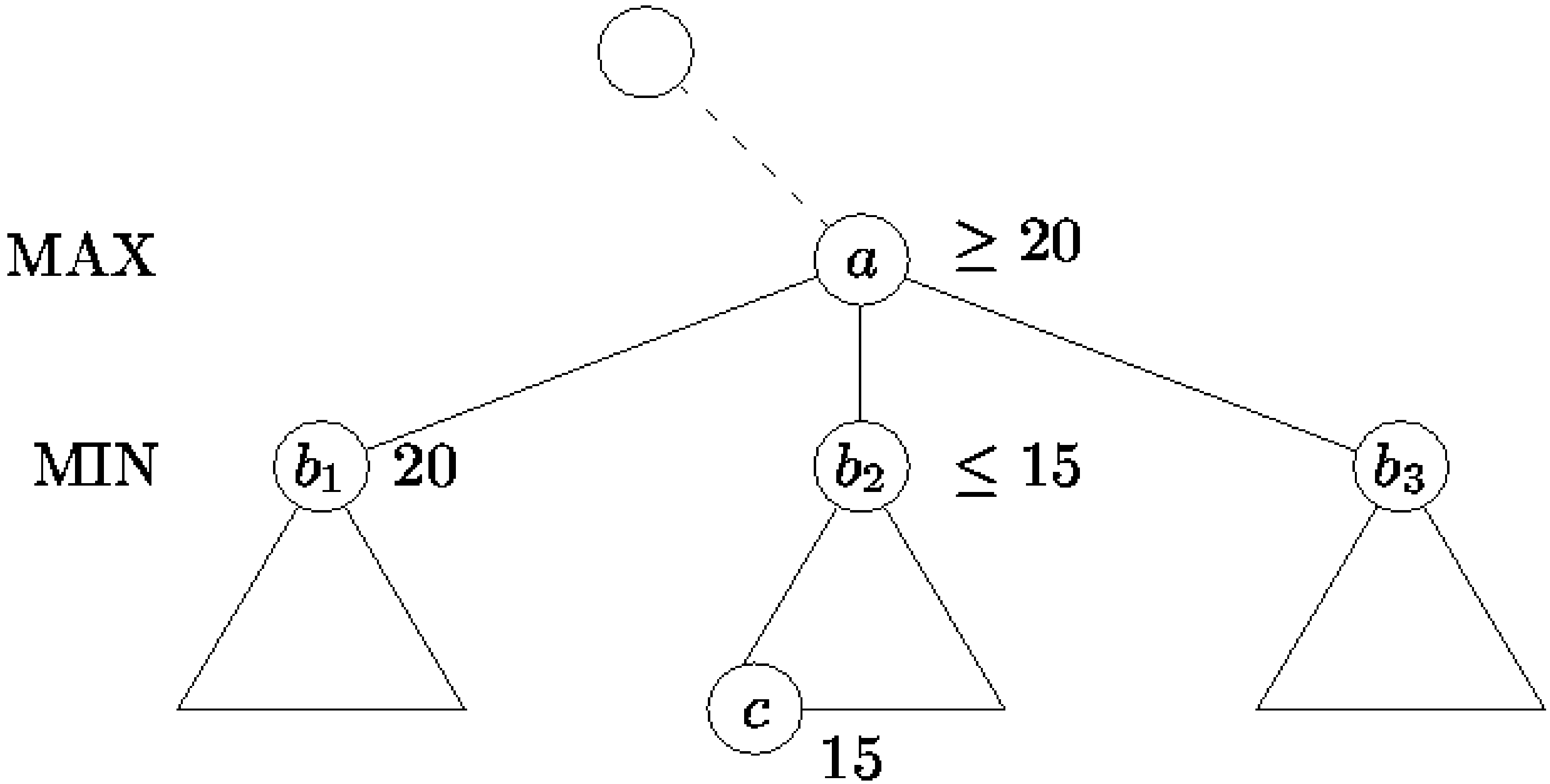


Στάθης Ζάχος
Άρης Παγουρτζής

Υλοποίηση Οπισθοδρόμησης

- Κατασκευή δέντρου
- Διάσχιση postorder
- Χρειάζεται εξοικονόμηση χώρου

A-B Prunning



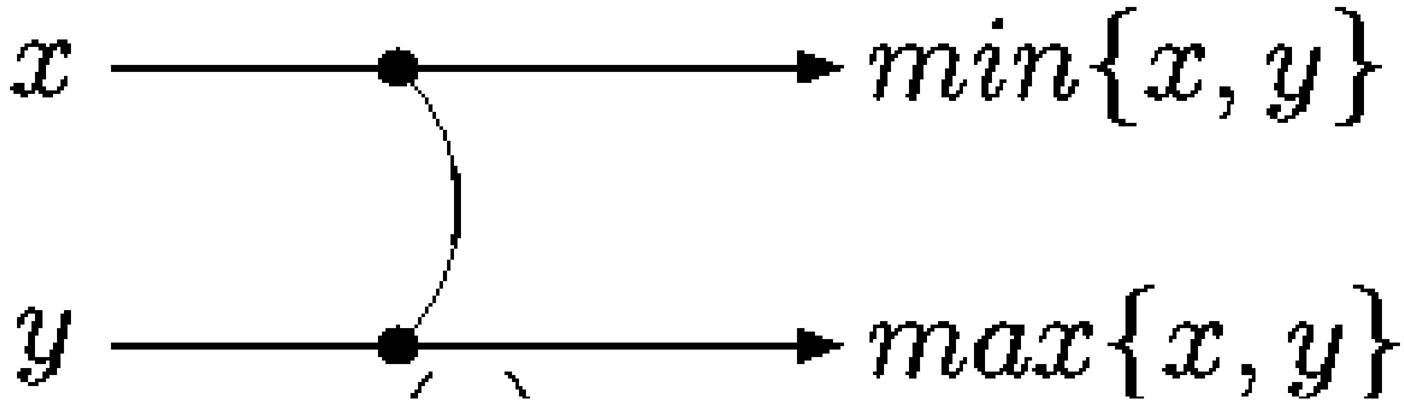
Branch and Bound

- Με χρήση μεθόδων εκτίμησης ορίων για κάθε κόμβο.
- Αν η μέχρι στιγμής λύση είναι μικρότερη από το κάτω όριο ενός κόμβου τότε δε χρειάζεται να εξετάσουμε τον κόμβο (σε περίπτωση που θέλουμε ελαχιστοποίηση).
- Αντίστοιχα, αν η τρέχουσα λύση είναι μεγαλύτερη από το άνω όριο (σε περίπτωση μεγιστοποίησης).

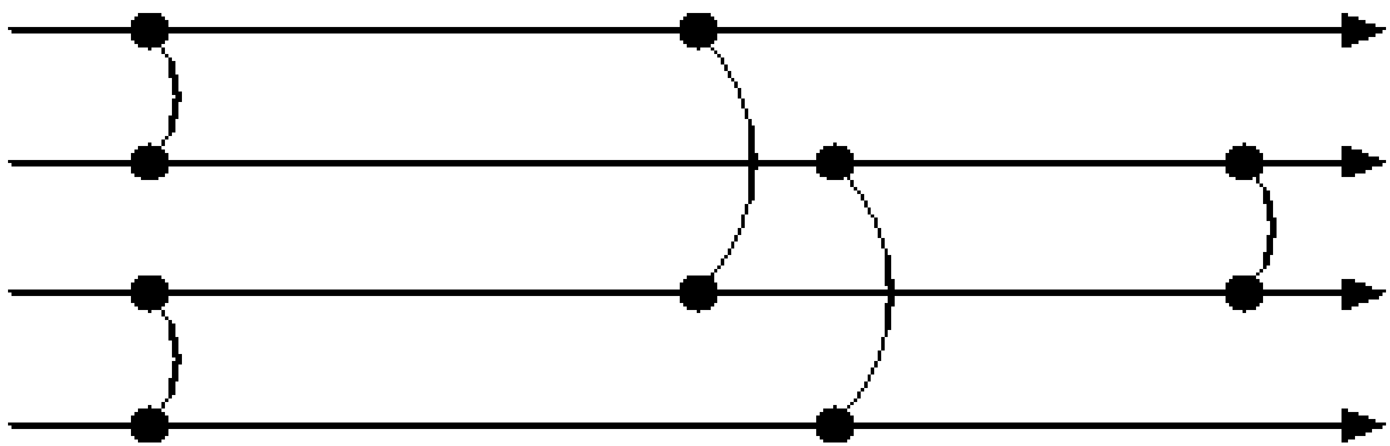
Δίκτυα Ταξινόμησης

- Ταξινόμηση ακολουθίας αριθμών με συγκρίσεις
- Επιτρέπονται ταυτόχρονες συγκρίσεις (παραλληλία) ζευγών διαφορετικών αριθμών
- Παριστάνονται με οριζόντιες γραμμές, 1 για κάθε είσοδο
- Έξοδος ταξινομημένη από «πάνω» προς τα «κάτω»

Συγκριτής (Comparator)



Δίκτυο Ταξινόμησης 4 εισόδων

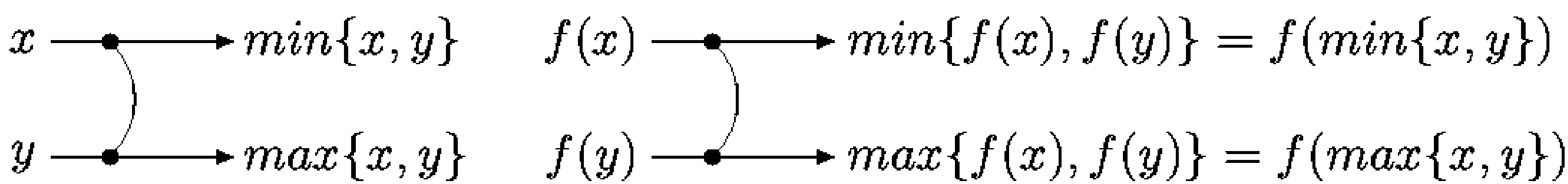


Βάθος δικτύου: 3   Μέγεθος δικτύου: 5

Αρχή 0-1

Αν ένα δίκτυο ταξινομεί οποιαδήποτε δυαδική ακολουθία, τότε ταξινομεί οποιαδήποτε ακολουθία

Βασίζεται στην ιδιότητα διατήρησης της διάταξης κατά την εφαρμογή αύξουσας συνάρτησης *f* σε εισόδους και εξόδους:



Αρχή 0-1

<u>Απόδειξη:</u>

Έστω $<a_1, a_2, \ldots , a_n>$ που δεν ταξινομείται σωστά, π.χ. $a_\kappa > a_i$ ενώ το a_κ εμφανίζεται πάνω από το a_i στην έξοδο.

Χρησιμοποιώντας:

$$f(x) = \begin{cases} 0 & x \leq a_i \\ 1 & x > a_i \end{cases}$$

σε εισόδους και εξόδους προκύπτει ότι ούτε η δυαδική $<f(a_1), f(a_2), \ldots , f(a_n)>$ ταξινομείται.

Bitonic ακολουθίες

- Είναι της μορφής 0*1*0* ή 1*0*1*

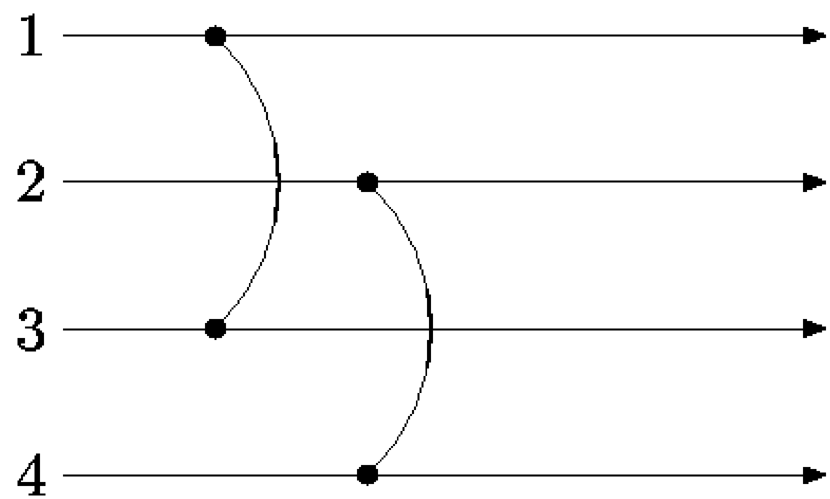
- Προκύπτουν από 2 ταξινομημένες δυαδικές ακολουθίες ενωμένες σε σειρά:

  αύξουσα - φθίνουσα

  ή

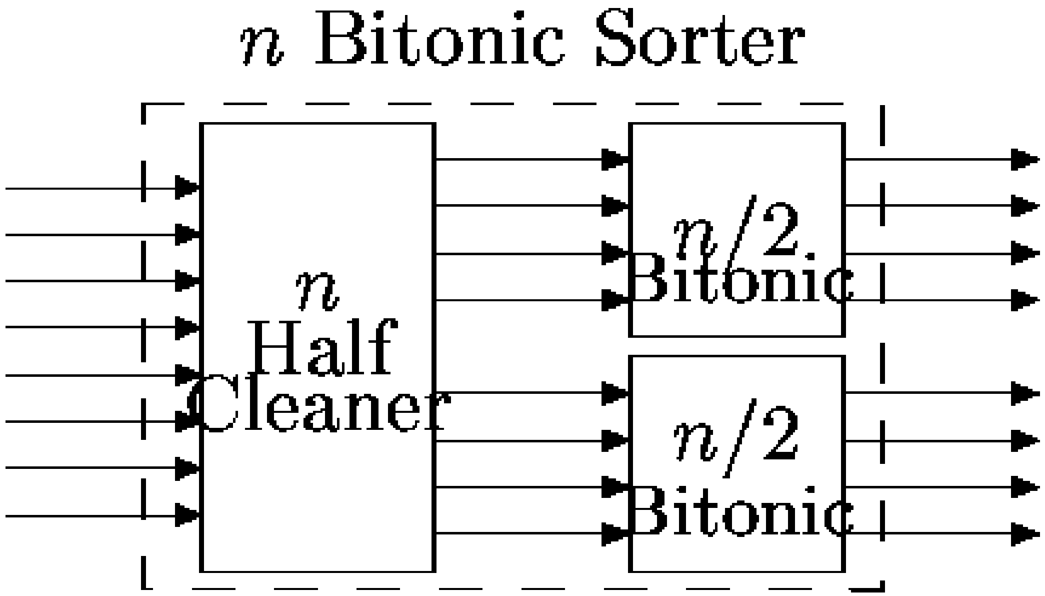
  φθίνουσα - αύξουσα

Δίκτυο Half Cleaner

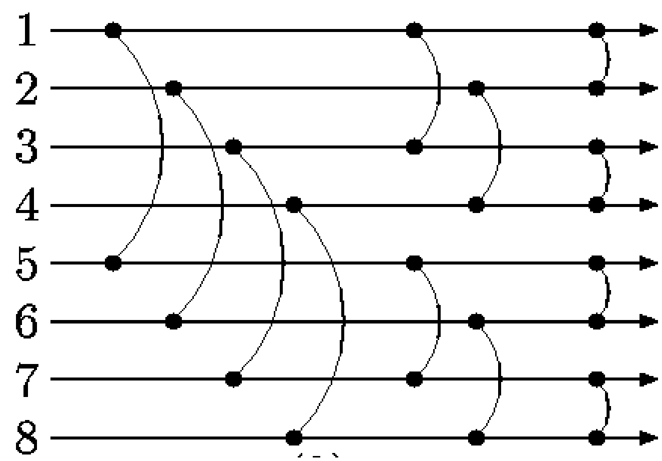


- 2κ γραμμές, κ παράλληλοι συγκριτές
- Κάθε συγκριτής συνδέει γραμμές i, i+k
- Με είσοδο bitonic στην έξοδο:
  κάθε μισό είναι bitonic και
  1ο μισό <0,0,...,0> ή 2ο μισό <1,1,...,1>

Ταξινόμηση Bitonic Ακολουθιών

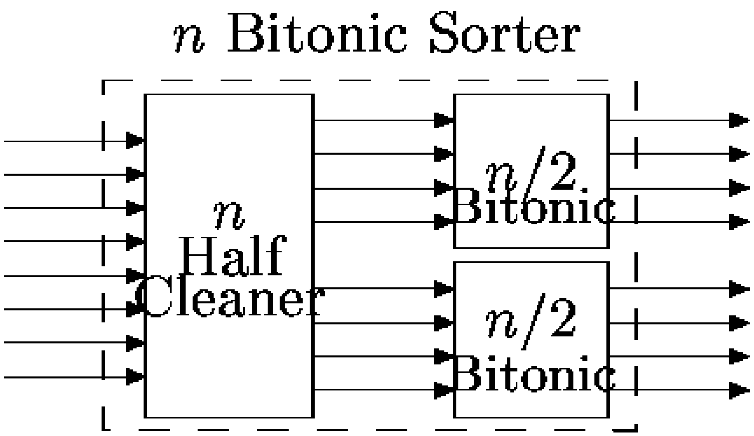


Για 8 εισόδους:



Στάθης Ζάχος
Άρης Παγουρτζής

Βάθος και Μέγεθος Bitonic Sorter



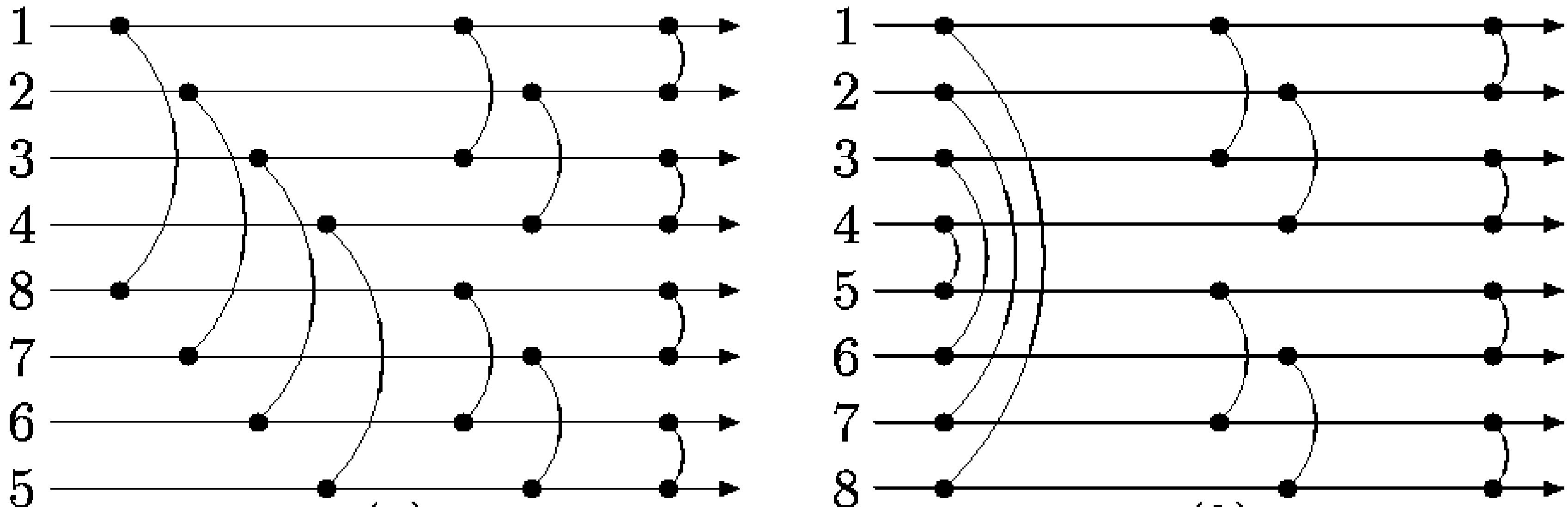
Βάθος

$$\begin{aligned} D(n) &= 1 + D(\frac{n}{2}) \\ &= \log_2 n \end{aligned}$$

Μέγεθος

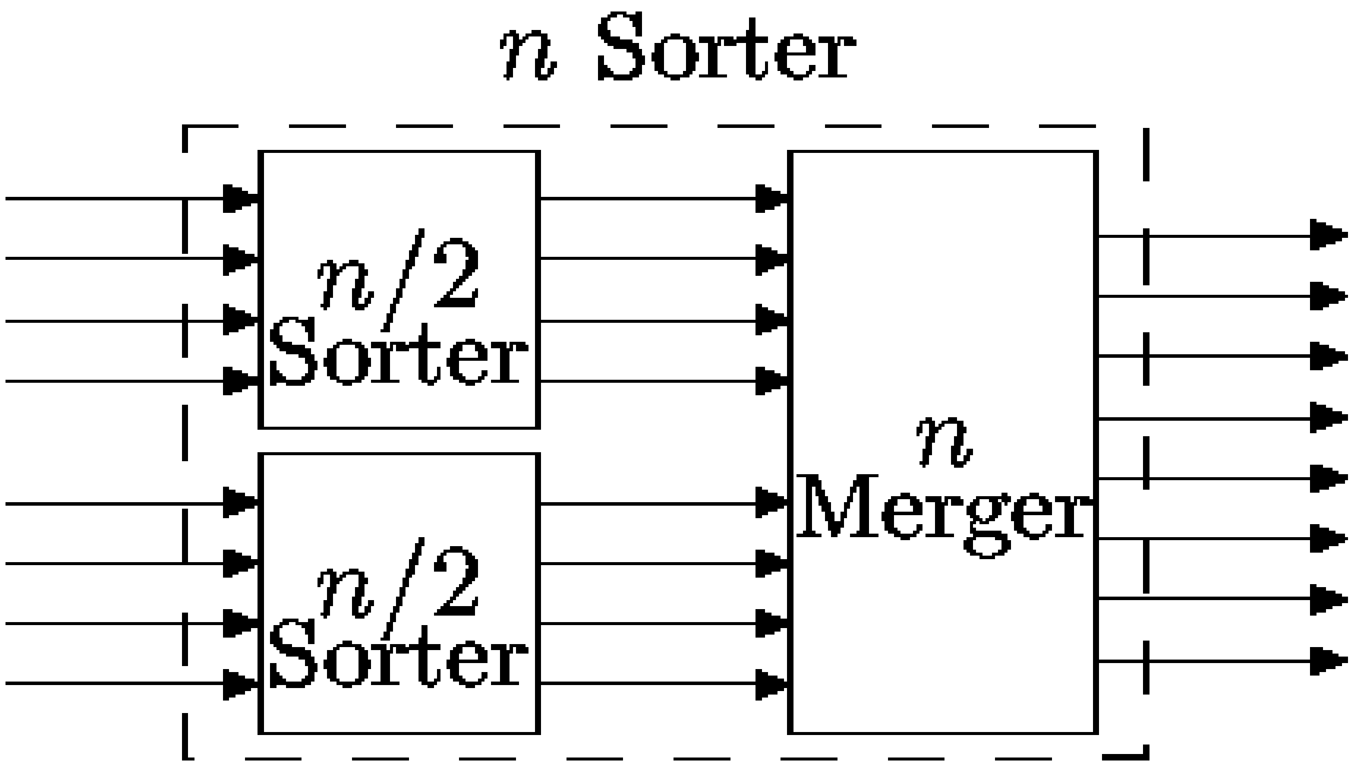
$$\begin{aligned} S(n) &= \frac{n}{2} + 2S(\frac{n}{2}) \\ &= \frac{n}{2} + 2\frac{n}{4} + 4\frac{n}{8} + \cdots + \frac{n}{2} \\ &= \frac{n}{2}\log_2 n \end{aligned}$$

Συγχωνευτής (Merger)

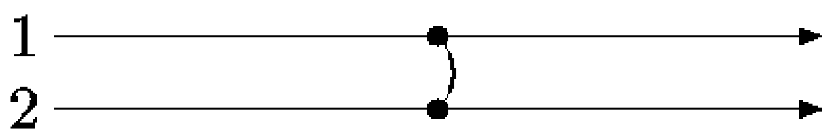


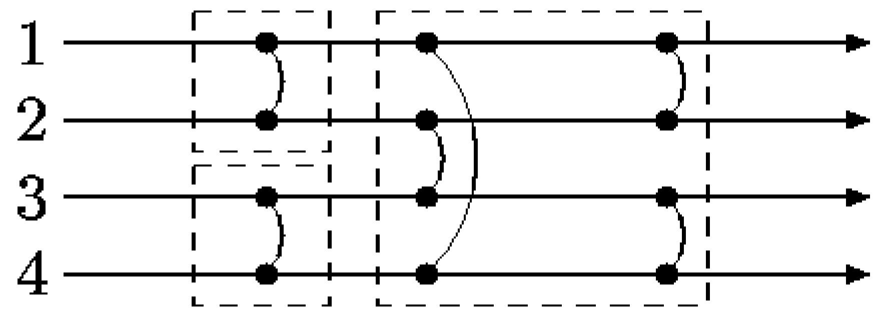
- Δέχεται δύο ταξινομημένες ακολουθίες μεγέθους n/2, παράγει 1 ταξινομημένη μεγέθους n.
- Προκύπτει από Bitonic Sorter με αντιστροφή του 2ου μισού

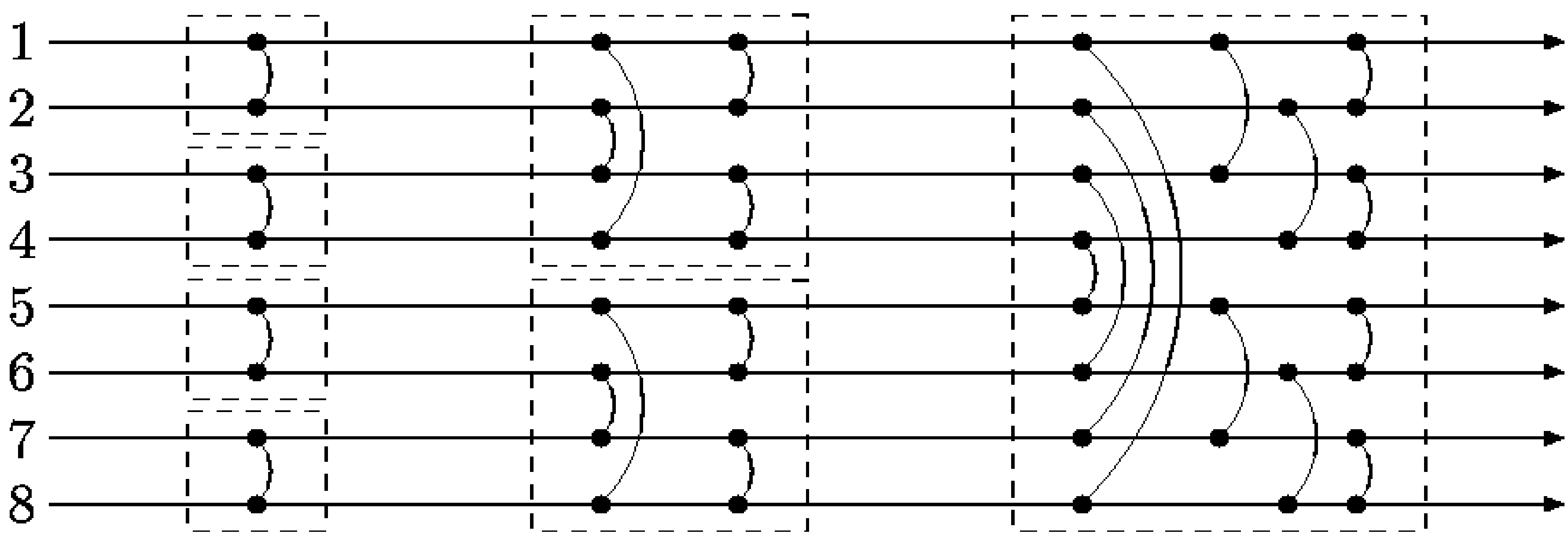
Δίκτυο Ταξινόμησης n εισόδων



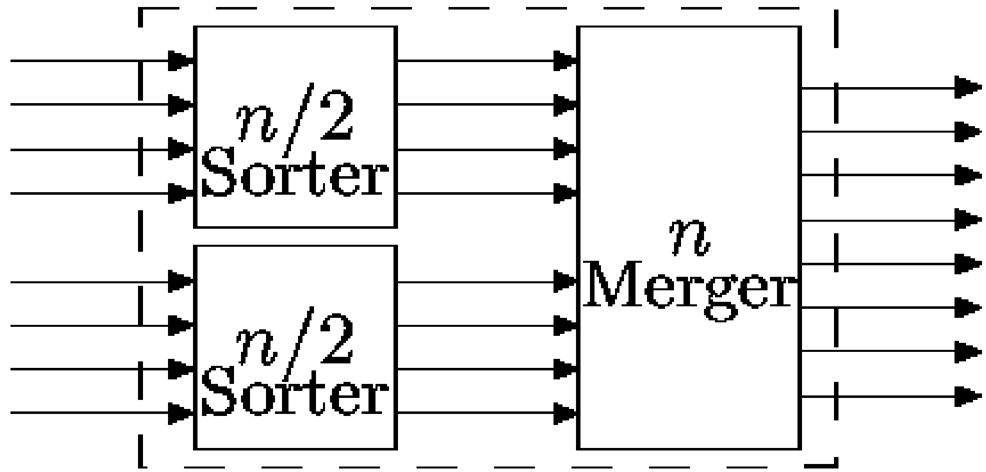
Sorter 2,4 και 8 εισόδων







Βάθος και Μέγεθος Sorter



$$D(n) = D(n/2) + \log n = \ldots = O(\log n)$$

$$S(n) = 2S(n/2) + n \log^2 n/2 = \ldots = O(n \log^2 n)$$