



Τεχνικές Σχεδίασης Αλγορίθμων

- Divide and Conquer (aesthetic!)
 - Merge Sort, QSort,...
- Greedy algorithms (elegance!)
 - Prim, Kruskal, Huffman,...
- Dynamic Programming (the hammer)
 - Bellman, LCS,...
- Backtracking (better than nothing)
 - κ-Queens, TSP,...



Divide And Conquer

- Διαίρεση του προβλήματος σε μικρότερα προβλήματα
- Επίλυση με τον ίδιο τρόπο των μικροτέρων προβλημάτων
- Συνδυασμός των λύσεων για την εύρεση της αρχικής λύσης



Διχοτομική Αναζήτηση

- Πίνακας με ονόματα σε αλφαβητική σειρά (όπως ο τηλεφωνικός κατάλογος)

- Αλγόριθμος (αναζητούμενο στοιχείο: **x**)

repeat

Σύγκριση του αναζητούμενου στοιχείου με το μεσαίο στοιχείο

Αν είναι μικρότερο τότε συνεχίζουμε την αναζήτηση στο πρώτο ήμισυ

διαφορετικά συνεχίζουμε στο δεύτερο

until (x=μεσαίο στοιχείο ή πίνακας κενός)



Διχοτομική Αναζήτηση

- Πολυπλοκότητα \rightarrow Πίνακας ταξινομημένος, $T(n)$ η πολυπλοκότητα
- $T(n) = \begin{cases} a \text{ αν } n=1 \\ T(\frac{n}{2})+c \text{ αν } n>1 \end{cases} \quad \left| \quad a, c \text{ σταθερές} \right.$
- Αν $n = 2^k$ (πλήθος στοιχείων) τότε $T_n = T_1 + c \log n$
δηλαδή $\Theta(\log n)$



Mergesort

- **Διαίρεση** του πίνακα σε δύο υπο-πίνακες
- **Συγχώνευση** των δύο ταξινομημένων υπο-πινάκων
- Χρησιμοποίηση ενός δεύτερου πίνακα ίσου με τον αρχικό
- Πολυπλοκότητα:

$$T_n = \begin{cases} 0 & \text{αν } n=0 \\ 1 & \text{αν } n=1 \\ n+2T_{\frac{n}{2}} & \text{αν } n \geq 2 \end{cases}$$

Επομένως $T_n = O(n \log n)$



Άπληστοι αλγόριθμοι (Greedy Algorithms)

Ένας άπληστος αλγόριθμος πάντα κάνει την επιλογή που φαίνεται καλύτερη τη δεδομένη στιγμή



Άπληστοι αλγόριθμοι

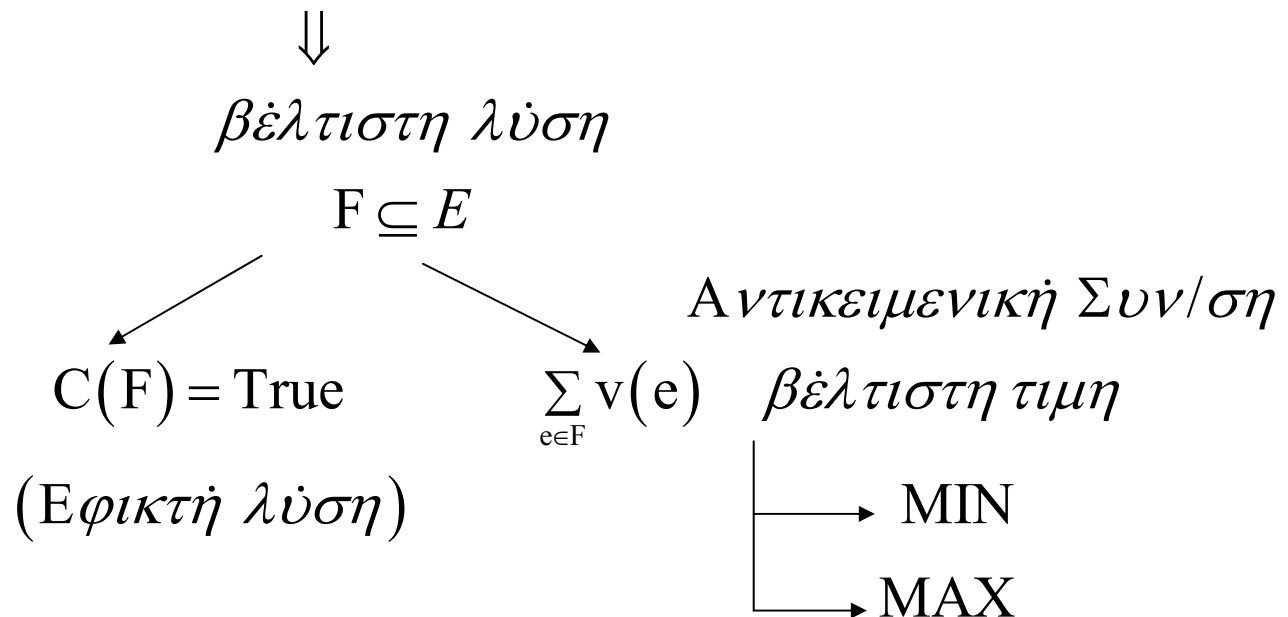
Ελπίδα: τοπικά βέλτιστη επιλογή οδηγεί σε μια ολικά βέλτιστη λύση

ΝΑΙ: για μερικά προβλήματα

ΌΧΙ: για κάποια άλλα

Προβλήματα

- E πεπερασμένο
- $\forall e \in E \rightarrow v(e) \in \mathcal{N}$ (Αντικειμενική Συν/ση)
- $C : P(E) \rightarrow \{\text{True}, \text{False}\}$ (περιορισμοί)





Γενικός αλγόριθμος

Greedy (E)

$S := \emptyset$ {solution}

for all elements of E do

$\left(\begin{array}{l} \text{Επέλεξε } x \in E; \\ E := E - \{x\} \\ \text{if } S \cup \{x\} \text{ feasible} \\ \text{then} \\ \quad S := S \cup \{x\} \\ \quad \text{Ενημέρωση Αντικειμενικής Συναρτ.} \end{array} \right.$

Greedy := S

End



Αλγόριθμος Kruskal

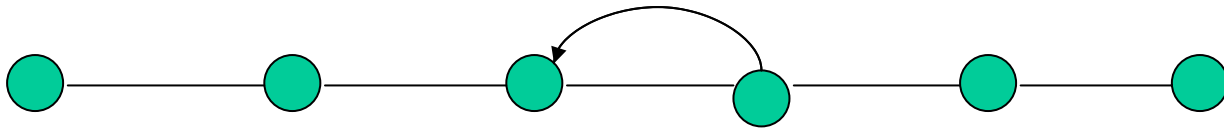
```
T ← ∅  
while (|T| < n-1) and (E ≠ ∅) do  
  begin  
    e ← smallest edge in E;  
    E ← E - {e};  
    if (T ∪ {e} has no cycle) then  
      T ← T ∪ {e}  
  end;  
if (|T| < n-1) then  
  write "network disconnected";
```

Πολυπλκότητα: $O(m \log n)$

Cycle: $O(\log n)$

Δυναμικός Προγραμματισμός

- **Βασική Αρχή:** κάθε υπό-στρατηγική μιας βέλτιστης στρατηγικής είναι η ίδια βέλτιστη



- **Μέθοδος:**
 - ένταξη του προβλήματος σε μια οικογένεια προβλημάτων της ίδιας φύσης
 - σύνδεση των βέλτιστων λύσεων με μια αναδρομική σχέση

Βασική προσέγγιση

Πως;

- Ορισμός υπο-προβλημάτων
- Σύνδεση της βέλτιστης λύσης με τις βέλτιστες λύσεις μέσω μιας αναδρομικής σχέσης
- Κατασκευή ενός πίνακα από λυμένα υπο-προβλήματα που χρησιμοποιούνται για την επίλυση των μεγαλύτερων
- Εύρεση της τιμής της λύσης από τα κάτω προς τα πάνω
- Κατασκευή της δομής της λύσης από πάνω προς τα κάτω

Στο γράφο $G = (X, U)$ $|X| = n$

ή $(G = (X, \Gamma))$

n - προβλήματα

$1 \longrightarrow 1$ ($P(1)$)

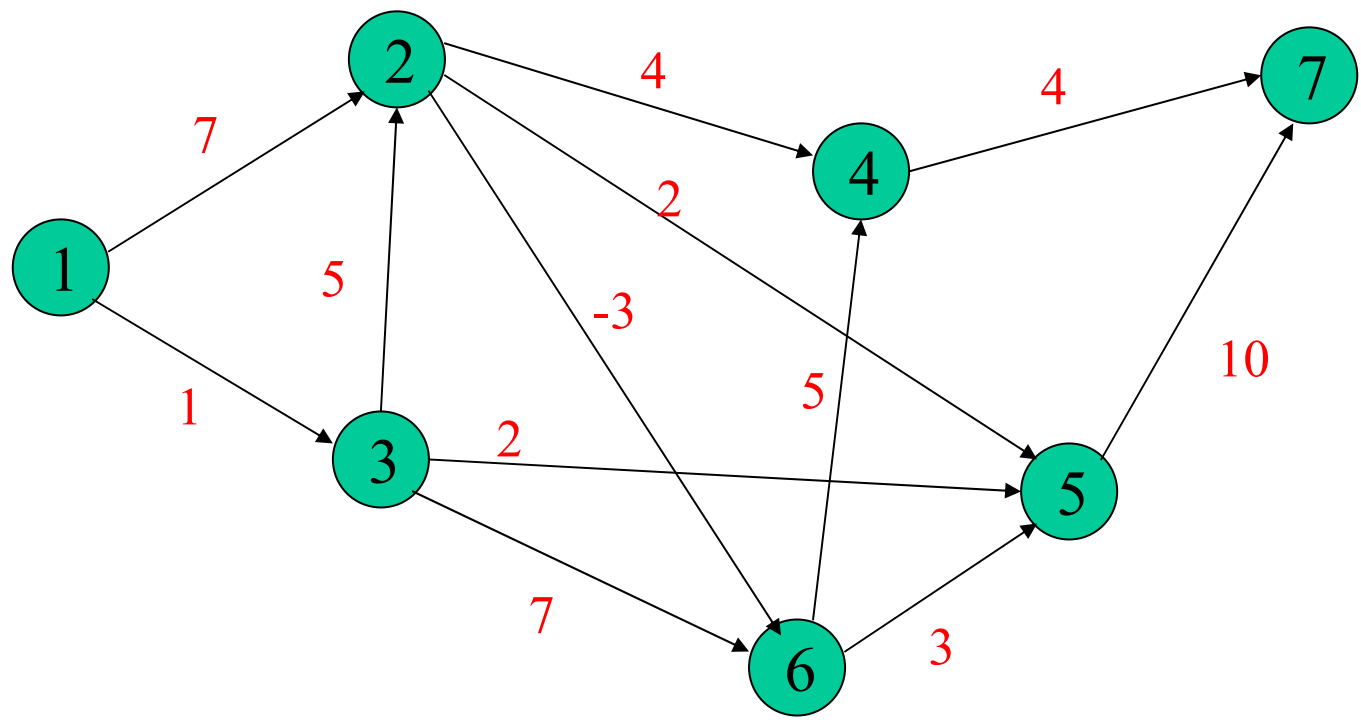
$1 \longrightarrow 2$ ($P(2)$)

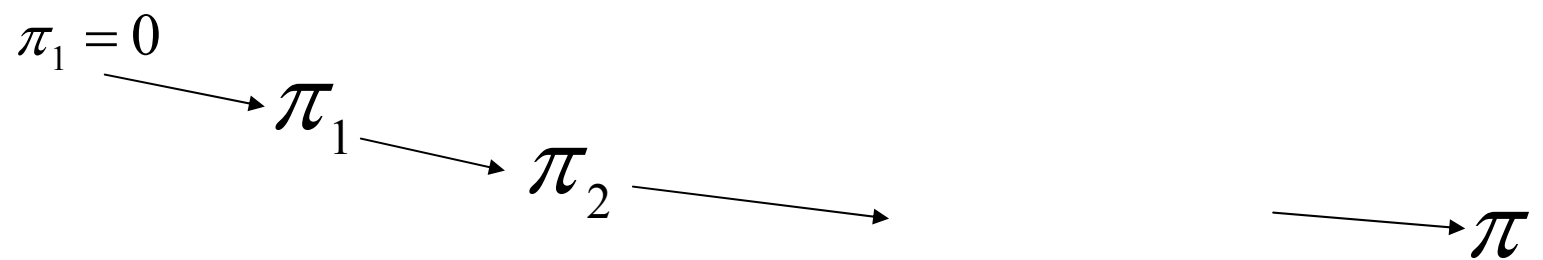
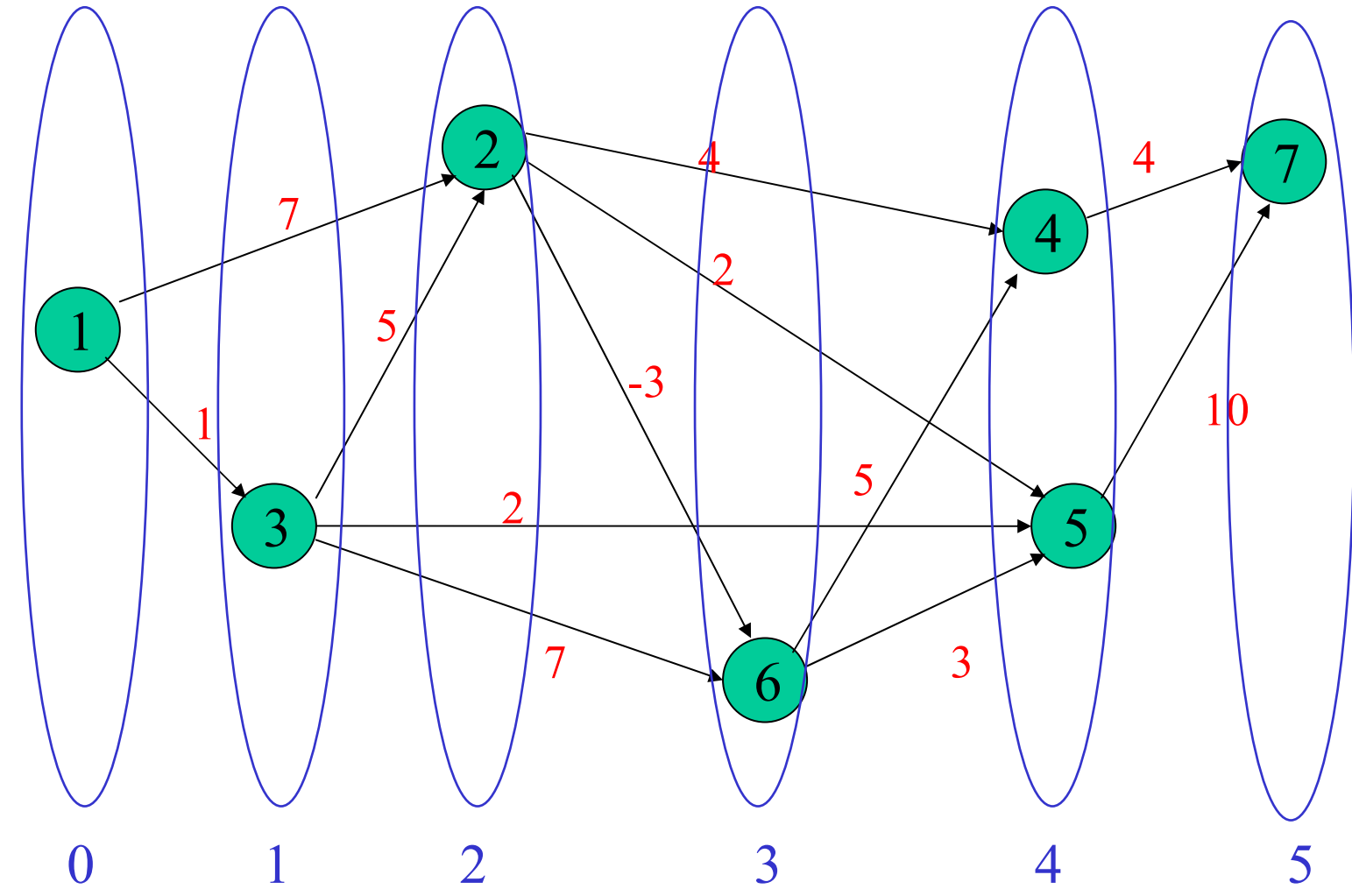
\vdots

$1 \longrightarrow n$ ($P(n)$)

$P(i): 1 \longrightarrow i, V(i)$ βέλτιστη τιμή λύσης

$$V(i) = \min_{j \in \Gamma^{-1}(i)} \{ V(j) + w_{ji} \}$$





γράφος $G = (X, U)$ $|X| = n$

$$V_k(i) = \{ \text{ΣΜ "1} \longrightarrow i" \mid \text{το πολύ } k \text{ τόξα} \}$$

$$V_k(i) = \min_{j \in \Gamma^{-1}(i)} \{ V_{k-1}(j) + w_{ji}, V_{k-1}(i) \}$$

$$V^*(i) = V_{n-1}(i) \quad (\text{απλά μονοπάτια})$$

Προβλήματα

$$V_0(i) = 0, +\infty \quad i = 1, 2, 3, \dots, n$$

$$V_1(i) = +\infty \quad \text{αν } i \notin \Gamma(1)$$

$$V_2(i)$$

$$V_3(i)$$

⋮

$$V_{n-1}(i) \quad (\text{Βέλτιστη τιμή})$$

Λύση

$$P^k(i) = j \quad \text{αν } V_k(i) = V_{k-1}(j) + w_{ji}$$

Πλήρης Αναζήτηση (Exhaustive search)

Πλήρης αναζήτηση = Απαρίθμηση όλων των λύσεων!
Better than nothing!

Πλήρης Αναζήτηση (Exhaustive search)

- K-Queens
- TSP
- Permutations
 - Διαχώρισης και αποτίμησης (B&B)
 - Γραμμικός Προγραμματισμός (Linear Programming, the axe!)
 - Προσεγγιστικοί αλγόριθμοι (Approximation Algorithms)
 - Ευριστικοί – μεταευριστικοί (Heuristics - Metaheuristics)