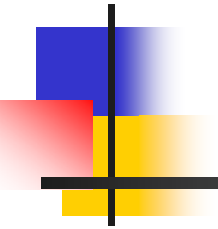
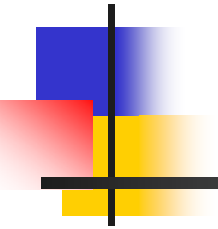


Άπληστοι αλγόριθμοι (Greedy Algorithms)

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*
- $\forall\ e \in E \rightarrow v(e) \in \aleph$ $(\text{Αντικειμενικη } \Sigma\upsilon\nu/\sigma\eta)$
- $C : P(E) \rightarrow \{\text{True}, \text{False}\}$ $(\pi\varepsilon\rho\iota o\rho\iota\sigma\mu o\acute{\iota})$

$\Downarrow$

βέλτιστη λύση

$F \subseteq E$

$C(F) = \text{True}$

$(\text{Εφικτή } \lambda\acute{\upsilon}\sigma\eta)$

Αντικειμενική Συν/ση

$\sum_{e \in F} v(e)$ *βέλτιστη τιμη*

- MIN
- MAX

Άπληστος αλγόριθμος (greedy)

$\Rightarrow$ Σειρά (κανόνας)

$\Rightarrow$ Αρχική λύση $F=\varnothing$

$\Rightarrow$ Τοπική Επιλογή : $F=F\cup\{e\}$

(προοδευτ. αύξηση + ικανοπ. περιορισμών)

FF,NF,BF (διαχ. μνήμης)

FIFO,LRU,OPT (paging)

Γενικός αλγόριθμος

Greedy (E)

S := ∅ {solution}

for all elements of E do

Επέλεξε x ∈ E;

E := E-{x}

if S ∪ {x} feasible

then

S := S ∪ {x}

Ενημέρωση Αντικειμενικης Συναρτ.

Greedy := S

End

Παράδειγμα: Αλγόριθμος Kruskal

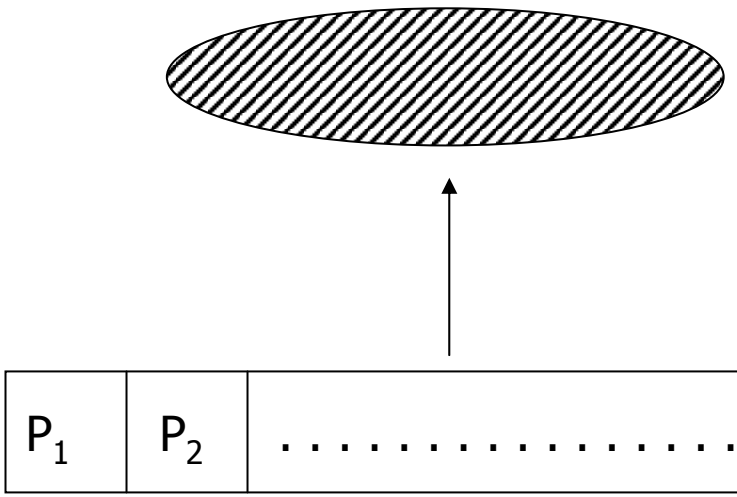
```
T ← ∅
while (|T| < n-1) and (E ≠ ∅) do

        e ← smallest edge in E;
        E ← E - {e};
        if (T ∪ {e} has no cycle) then
            T ← T ∪ {e}

if (|T| < n-1) then
    write "network disconnected";
```

Πολυπλκότητα: O(mlogn)
Cycle:O(logn)

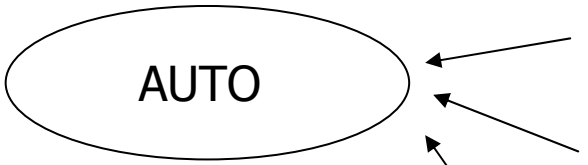
Διαχείριση πόρων



διάστημα χρησιμοπ.

t_i = (αρχή, τέλος)

Παράδειγμα



$e_1\ (a_1, T_1)$

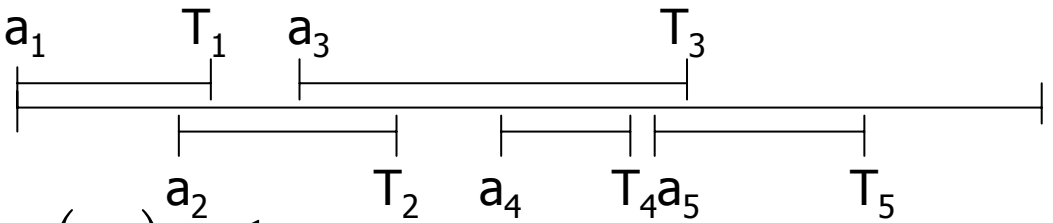
$e_2\ (a_2, T_2)$

$\vdots$

$e_n\ (a_n, T_n)$

1 μόνο πελάτης σε μία δεδομένη χρονική στιγμή

$\text{Ικανοποίηση Max Πελατών}$

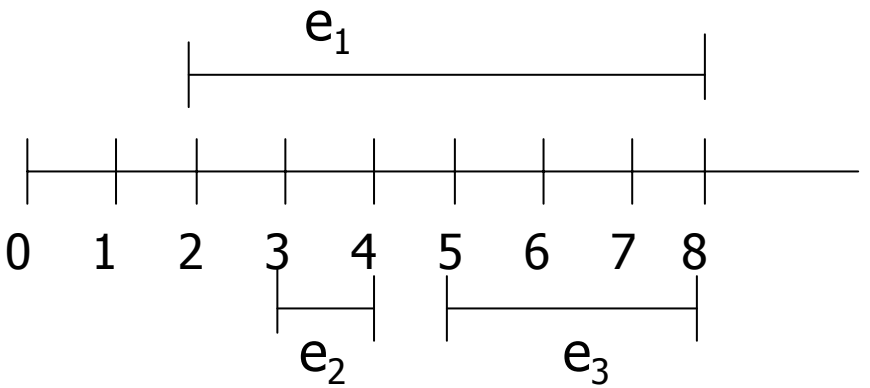


$v(e_i) = 1$

$F \subseteq E \,|\, \forall e_1, e_2 \in F \quad a(e_1) \le a(e_2) \Leftrightarrow T(e_1) \le a(e_2)$

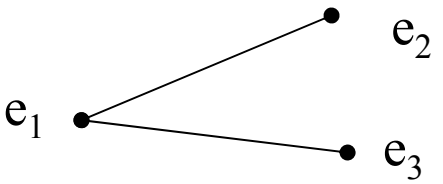
Παράδειγμα

	e_1	e_2	e_3
a	2	3	5
T	8	4	8



Τ α ξ / σ η : e_2 e_1 e_3 $\Rightarrow$ $|K| = 2$

4 8 8

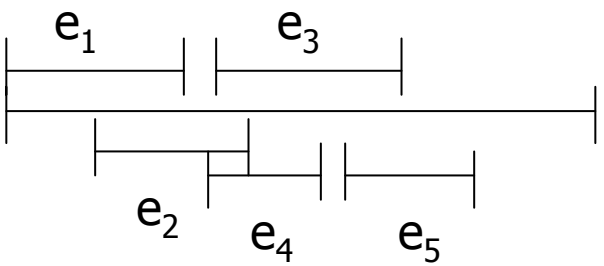


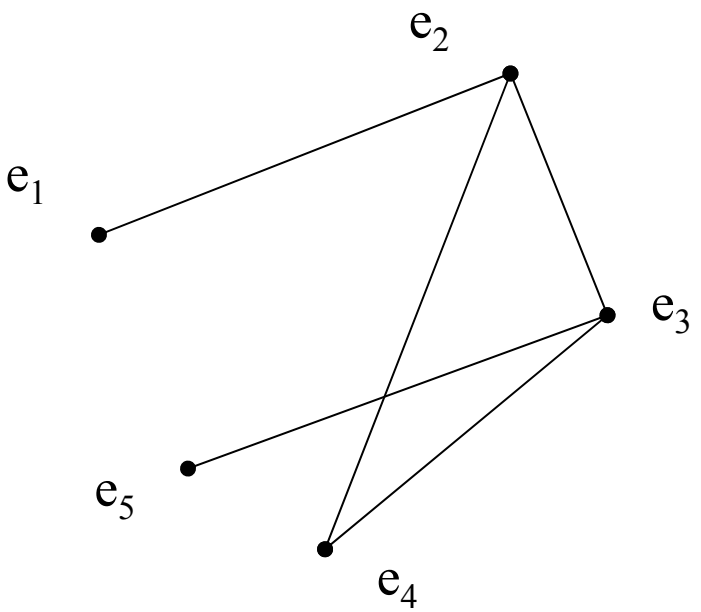
α ρ χ η ↗ e_1 e_2 e_3 $\Rightarrow$ $|F| = 1$

2 3 5

Φθίνουσα διάταξη

$\underline{\alpha\rho\chi\eta} \searrow \quad e_3 \quad e_2 \quad e_1 \quad \Rightarrow \quad |F| = 2$





Αλγόριθμος διαχείρισης ενός πόρου

Greedy AUTO

$\text{T}\alpha\xi / \sigma\varepsilon \;\; \text{e}_\text{i} : \text{T}(\text{e}_1) \le \text{T}(\text{e}_2) \le \cdots \le \text{T}(\text{e}_\text{n})$

$\text{F} = \varnothing$

for $\text{i} = 1$ to n

$\text{F} = \text{F} \cup \{\text{e}_\text{i}\}$ $\underline{\underline{\alpha\nu}}$

$$\left[\alpha(\text{e}_{\text{i-1}}), \text{T}(\text{e}_{\text{i-1}})\right] \cap \left[\alpha(\text{e}_\text{i}), \text{T}(\text{e}_\text{i})\right] = \varnothing$$

Απόδειξη βελτιστότητας

$$F = \{x_1, x_2, \ldots, x_k, \ldots, x_p\} \ \lambda\acute{\upsilon}\sigma\eta \ \text{Greedy}$$

$$Opt = \{y_1, y_2, \ldots, y_k, \ldots, y_q\} \ q \geq p \ \beta\grave{\varepsilon}\lambda\tau\iota\sigma\tau\eta$$

$$(T\alpha\xi/\sigma\eta \ \ \eta\mu\varepsilon\rho.\ \alpha\acute{\upsilon}\xi\upsilon\sigma\alpha)$$

$$x_1 = y_1, \ldots, x_k = y_k, x_{k+1} \neq y_{k+1}$$

$$F \ \alpha\pi o \ \kappa\alpha\tau\alpha\sigma\kappa\varepsilon\upsilon\grave{\eta} : T(x_{k+1}) \leq T(y_{k+!})$$

$$O'pt = \{x_1, x_2, \ldots, x_{k-1}, x_k, x_{k+1}, y_{k+2} \ldots, y_p\} \ldots$$

$$O'pt = \{x_1\, x_2, \ldots, x_{k-1}, x_k, x_{k+1}, \ldots x_p\}, p = q$$

Μεγιστοποίηση ολικής διάρκειας

Ολική διάρκεια ενοικίασης

AUTO : MAX ?

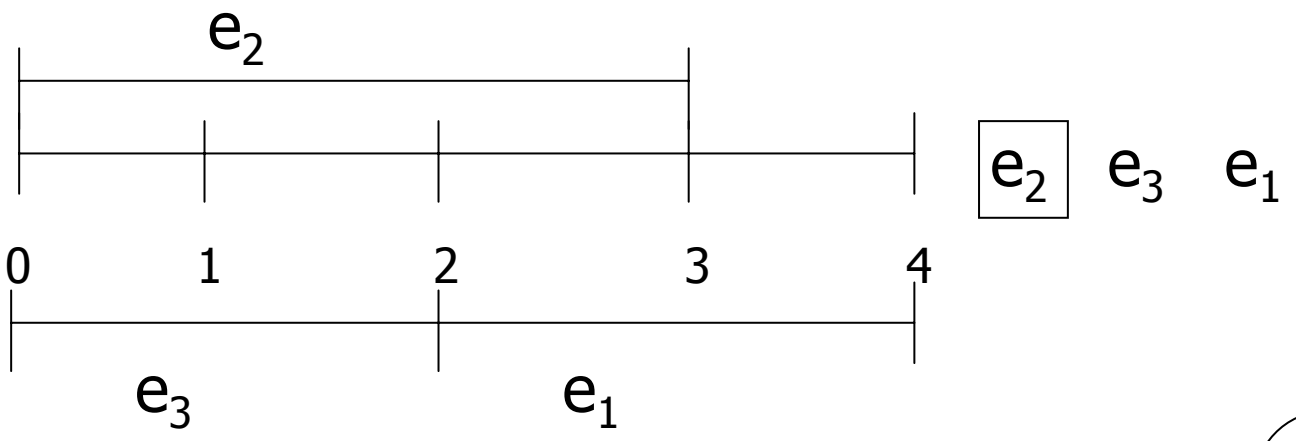
Greedy = ?

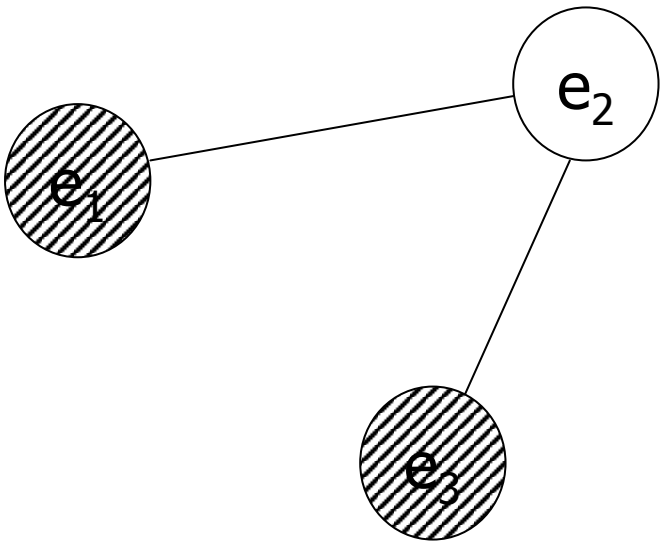
- Προτεραιότητα : μεγάλη διάρκεια

⇓

Ταξ/ση : διαρκειες ↘

Μοντελοποίηση με γράφους





Δύο Auto ?