



Άπληστοι αλγόριθμοι (Greedy Algorithms)

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



Άπληστοι αλγόριθμοι

Ελπίδα: τοπικά βέλτιστη επιλογή να οδηγεί σε μια ολικά βέλτιστη λύση

ΝΑΙ: για μερικά προβλήματα

ΌΧΙ: για κάποια άλλα

Προβλήματα

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



βέλτιστη λύση

$$F \subseteq E$$

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

(Εφικτή λύση)

Αντικειμενική Συν/ση

$$\sum_{e \in F} v(e) \quad \text{βέλτιστη τιμή}$$

MIN

MAX



Άπληστος αλγόριθμος (greedy)

⇒ Σειρά (κανόνας)

⇒ Αρχική λύση $F = \emptyset$

⇒ Τοπική Επιλογή : $F = F \cup \{e\}$

(προοδευτ. αύξηση + ικανοπ. περιορισμών)

FF, NF, BF (διαχ. μνήμης)

FIFO, LRU, OPT (paging)



Γενικός αλγόριθμος

Greedy (E)

$S := \emptyset$ {solution}

for all elements of E do

Επιλέξετε $x \in E$;

$E := E - \{x\}$

if $S \cup \{x\}$ feasible

then

$S := S \cup \{x\}$

Ενημέρωση Αντικειμενικής Συναρτ.

Greedy := S

End

Παράδειγμα: Αλγόριθμος Kruskal

$T \leftarrow \emptyset$

while ($|T| < n-1$) **and** ($E \neq \emptyset$) **do**

$e \leftarrow$ smallest edge in E ;

$E \leftarrow E - \{e\}$;

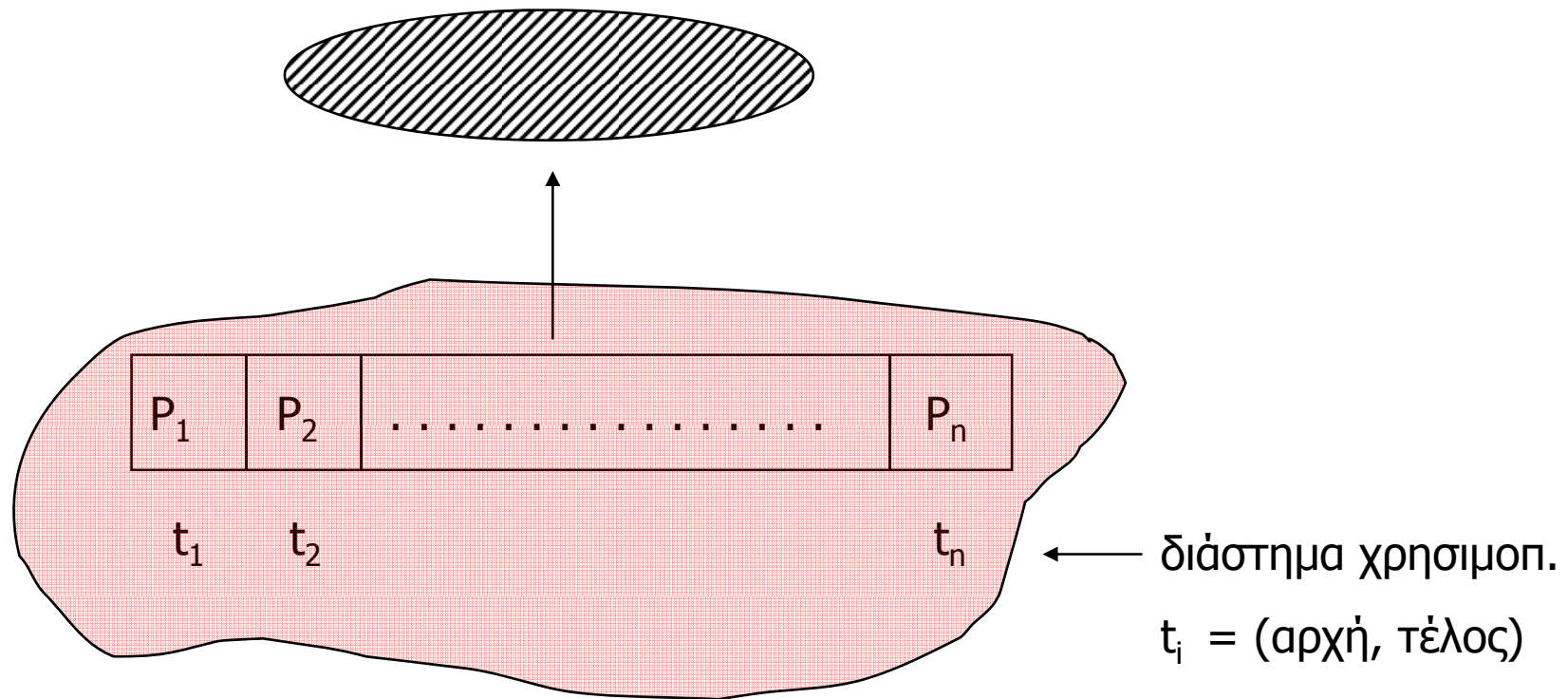
if ($T \cup \{e\}$ has no cycle)

then

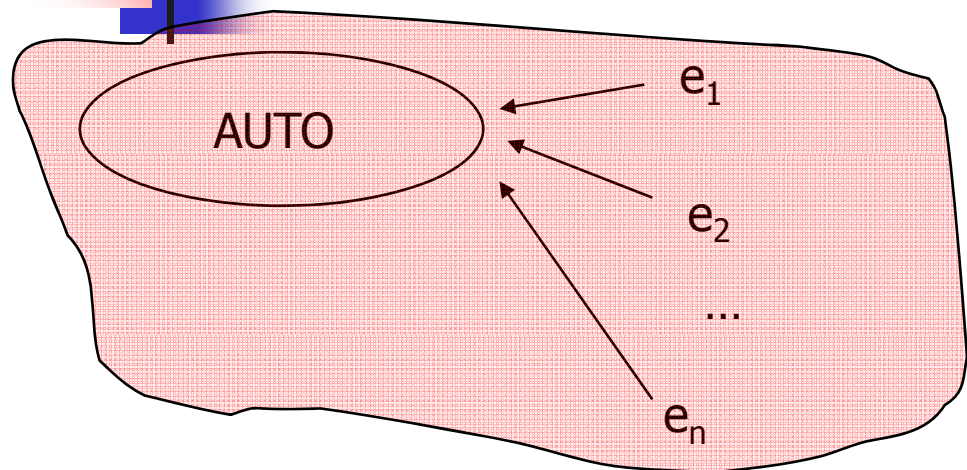
$T \leftarrow T \cup \{e\}$



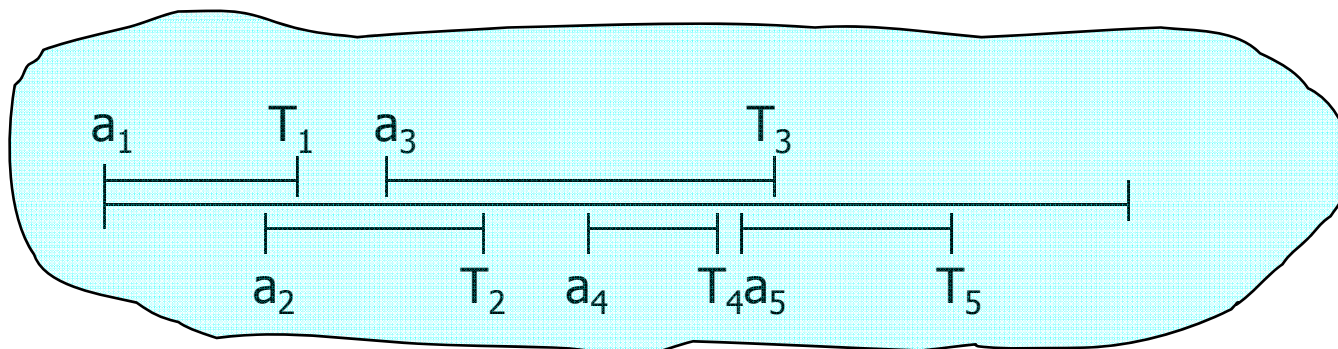
Διαχείριση πόρων



Παράδειγμα

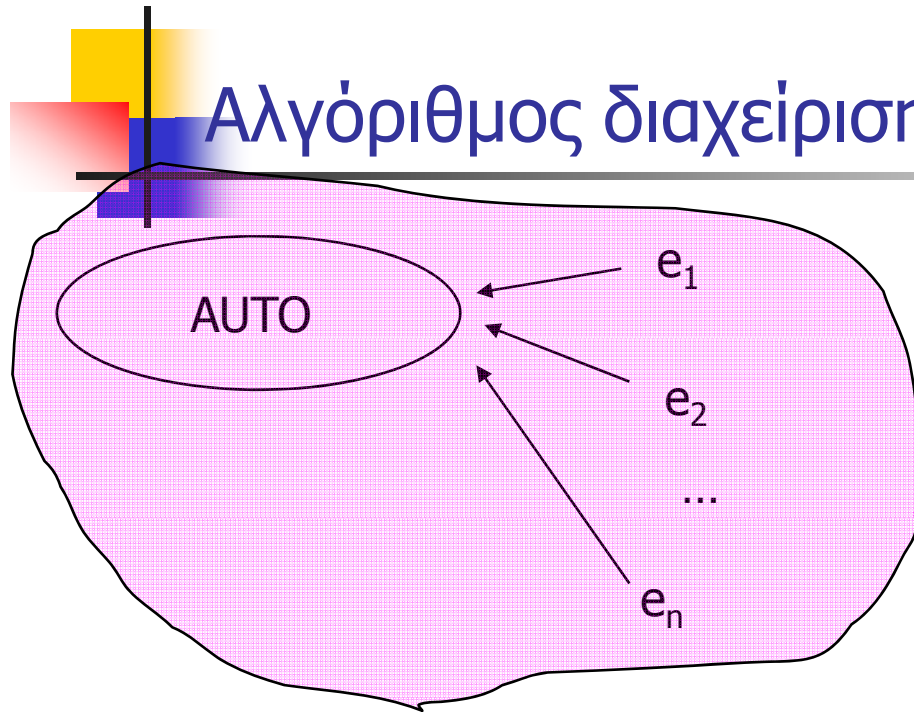


1 μόνο πελάτης σε μία δεδομένη χρονική στιγμή

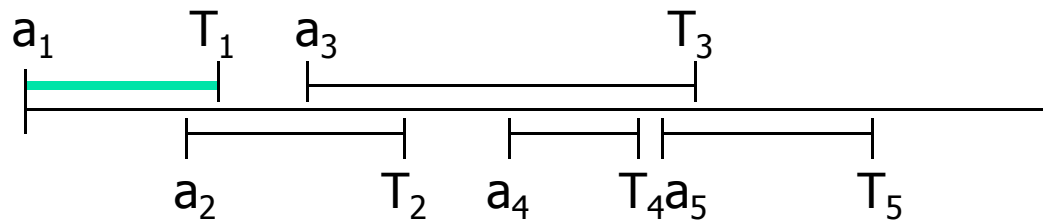


Στόχος: Ικανοποίηση μέγιστου αριθμού πελατών, $n=5$

Αλγόριθμος διαχείρισης ενός πόρου

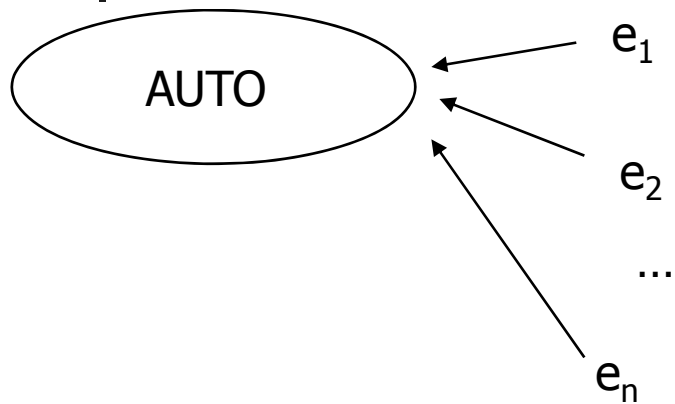


1 μόνο πελάτης σε μία δεδομένη χρονική στιγμή

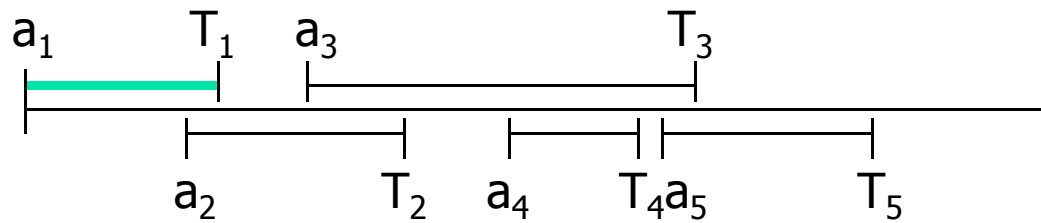


Βήμα 1

Αλγόριθμος διαχείρισης ενός πόρου

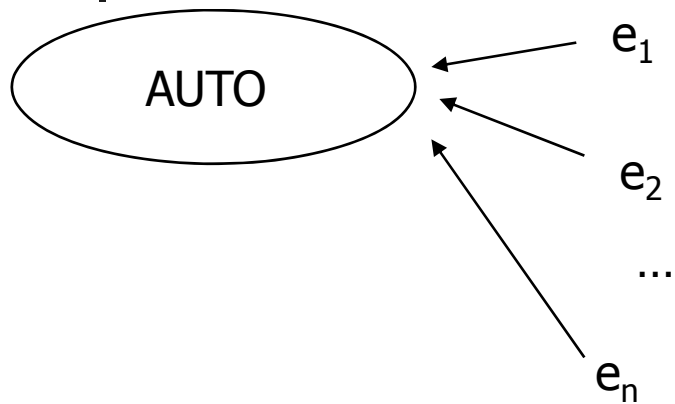


1 μόνο πελάτης σε μία
δεδομένη χρονική στιγμή

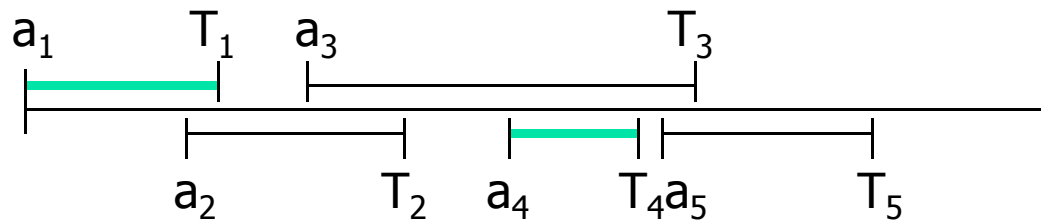


Βήμα 2 : e_2 απορρίπτεται

Αλγόριθμος διαχείρισης ενός πόρου

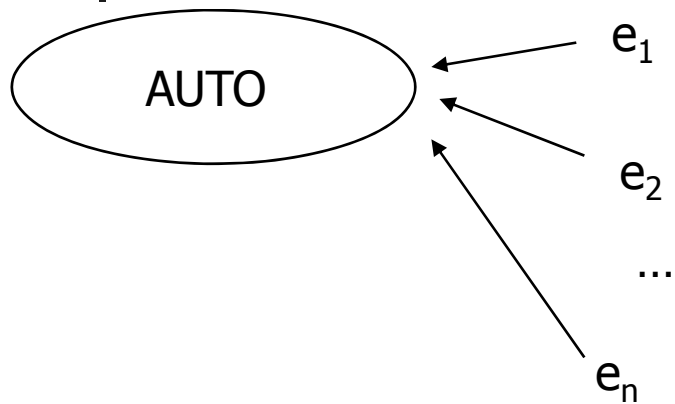


1 μόνο πελάτης σε μία
δεδομένη χρονική στιγμή

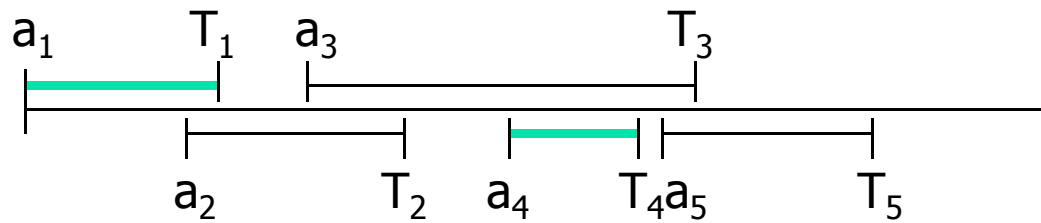


Βήμα 3

Αλγόριθμος διαχείρισης ενός πόρου

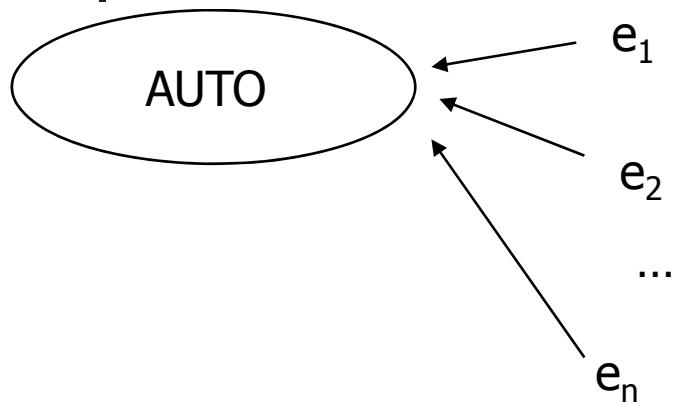


1 μόνο πελάτης σε μία
δεδομένη χρονική στιγμή

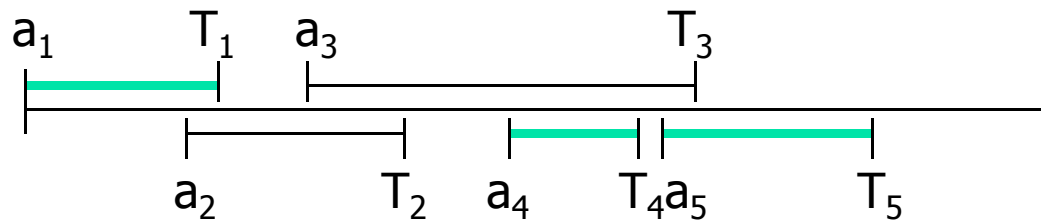


Βήμα 4 : e_3 απορρίπτεται

Αλγόριθμος διαχείρισης ενός πόρου



1 μόνο πελάτης σε μία
δεδομένη χρονική στιγμή



Βήμα 5 ($n=5$)



Αλγόριθμος διαχείρισης ενός πόρου

Greedy AUTO

Ταξινόμησε τα στοιχεία κατά φθίνουσα σειρά αποπεράτωσης

$\mathbf{F} \leftarrow \emptyset$

For $i=1$ to n **do**

if e_i και last in \mathbf{F} δεν τέμνονται **then**

$\mathbf{F} \leftarrow \mathbf{F} \cup \{e_i\};$



Απόδειξη βελτιστότητας

$F = \{x_1, x_2, \dots, x_k, \dots, x_p\}$ λύση Greedy

$Opt = \{y_1, y_2, \dots, y_k, \dots, y_q\}$ $q \geq p$ βέλτιστη

(Ταξίση ημερ. αύξουσα)

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

F απο κατασκευή : $T(x_{k+1}) \leq T(y_{k+1})$

$Opt = \{x_1, x_2, \dots, x_{k-1}, x_k, x_{k+1}, y_{k+2}, \dots, y_p\} \dots$

$Opt = \{x_1, x_2, \dots, x_{k-1}, x_k, x_{k+1}, \dots, x_p\}, p = q$



Μεγιστοποίηση ολικής διάρκειας

Ολική διάρκεια ενοικίασης
AUTO : MAX ?

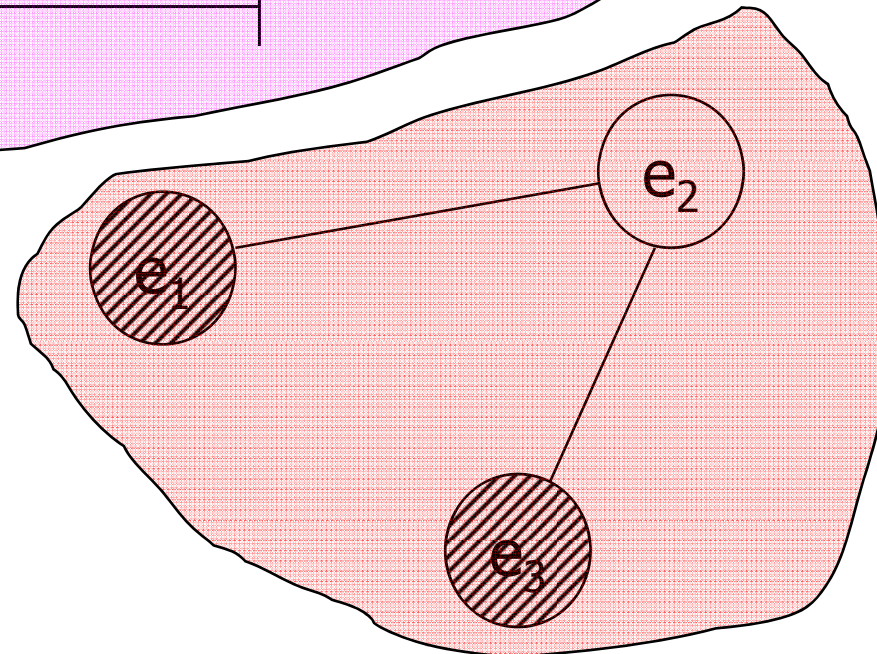
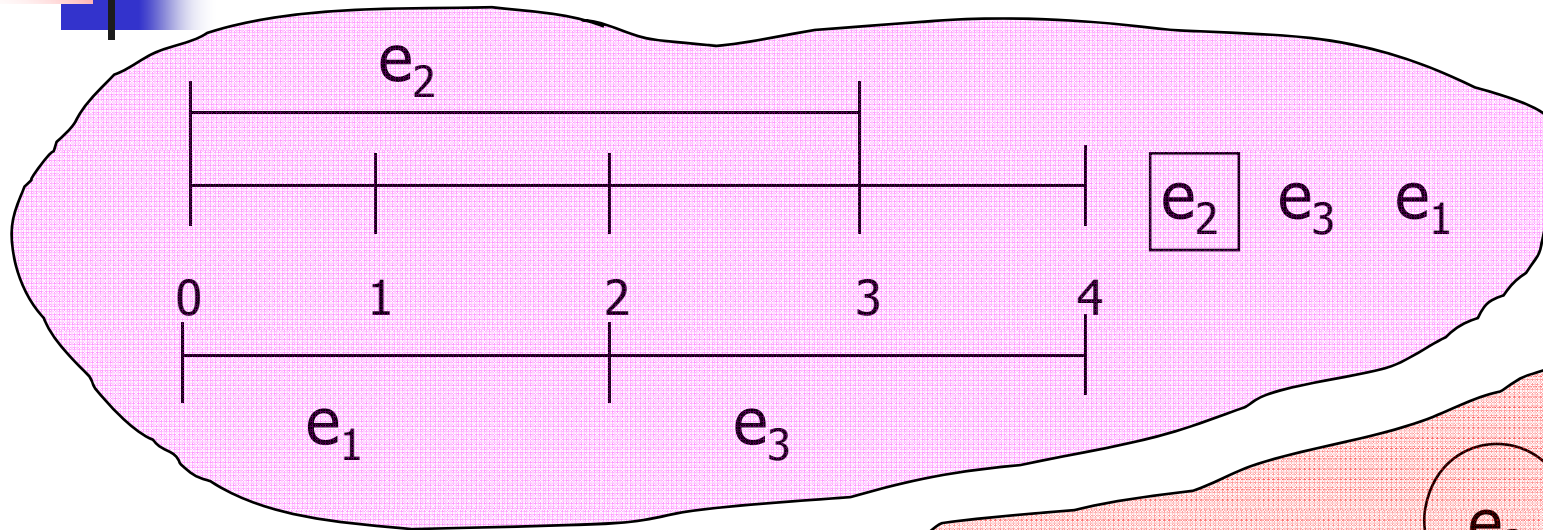
Greedy = ?

• Προτεραιότητα : μεγάλη διάρκεια

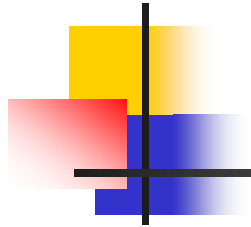


Ταξίση : διαρκείες

Μοντελοποίηση με γράφους



Δύο Auto ?



Level algorithms
