ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS
ΑΝΟΙΚΤΑ ακαδημαϊκά μαθήματα ΠΠ

Εισαγωγή στους Αλγορίθμους

Ενότητα 8η

Διδάσκων
Χρήστος Ζαρολιάγκης
Καθηγητής
Τμήμα Μηχανικών Η/Υ & Πληροφορικής
Πανεπιστήμιο Πατρών
Email: zaro@ceid.upatras.gr

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Άδειες Χρήσης

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.

- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Πατρών**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.

- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Σκοποί ενότητας

- Εισαγωγή στο «άπληστο» μοντέλο επίλυσης
- Περιγραφή και ανάλυση του αλγορίθμου χρονοπρογραμματισμού εργασιών

Περιεχόμενα ενότητας

- Χρονοπρογραμματισμός Διαστημάτων
- Χρονοπρογραμματισμός για Ελαχιστοποίηση Καθυστέρησης

Άπληστοι Αλγόριθμοι

- Χρονοπρογραμματισμός Διαστημάτων
- Χρονοπρογραμματισμός για Ελαχιστοποίηση Καθυστέρησης

Απληστία

> Greed is good. Greed is right. Greed works.
> Greed clarifies, cuts through, and captures the essence of the evolutionary spirit.
>
> *- Gordon Gecko (Michael Douglas)*

Εικόνα 1

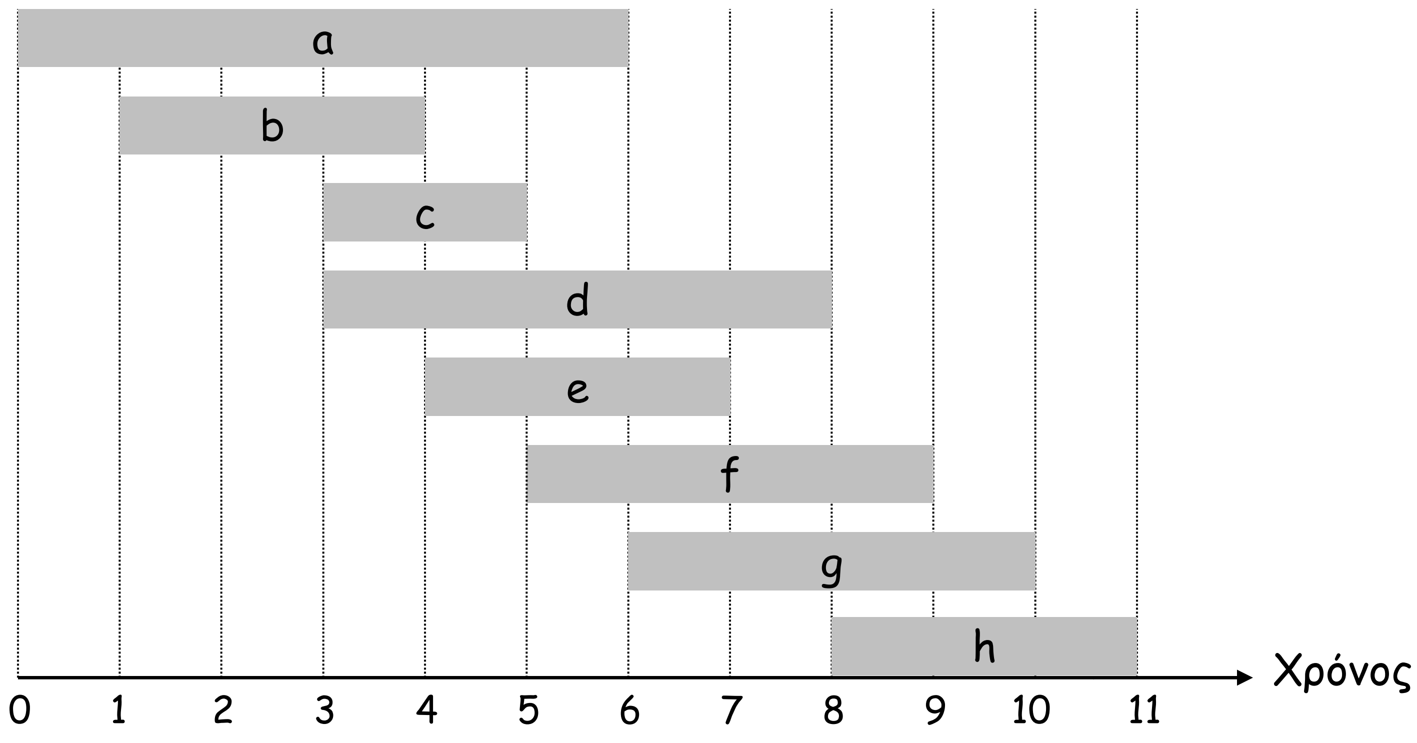
Απληστία. Δόμηση μιας λύσης σταδιακά, βελτιστοποιώντας μυωπικά κάποιο τοπικό κριτήριο σε κάθε βήμα

Χρονοπρογραμματισμός Διαστημάτων

Χρονοπρογραμματισμός Διαστημάτων

Χρονοπρογραμματισμός διαστημάτων.

- Η εργασία j ξεκινάει τη χρονική στιγμή s_j και τελειώνει τη χρονική στιγμή f_j
- Δύο εργασίες είναι συμβατές αν δεν επικαλύπτονται χρονικά
- Στόχος: εύρεση του μεγαλύτερου υποσυνόλου συμβατών μεταξύ τους εργασιών



Χρονοπρογραμματισμός Διαστημάτων – Άπληστοι Αλγόριθμοι

Άπληστο πρότυπο. Θεωρήστε τις εργασίες σε κάποια σειρά. Επιλέξτε κάθε εργασία εφόσον είναι συμβατή με αυτές που έχουν ήδη επιλεγεί.

- [Μικρότερος χρόνος έναρξης] Θεωρήστε τις εργασίες σε αύξουσα σειρά ως προς το χρόνο έναρξης s_j.
- [Μικρότερος χρόνος λήξης] Θεωρήστε τις εργασίες σε αύξουσα σειρά ως προς το χρόνο λήξης f_j.
- [Μικρότερο χρονικό διάστημα] Θεωρήστε τις εργασίες σε αύξουσα σειρά ως προς το χρονικό διάστημα $f_j - s_j$.
- [Ελάχιστες διενέξεις] Για κάθε εργασία, μετρήστε τον αριθμό των μη συμβατών εργασιών c_j. Χρονοπρογραμματίστε κατά αύξουσα σειρά διενέξεων c_j.

Άπληστο πρότυπο. Θεωρήστε τις εργασίες σε κάποια σειρά. Επιλέξτε κάθε εργασία εφόσον είναι συμβατή με αυτές που έχουν ήδη επιλεγεί

Δεν αποδίδει ο μικρότερος χρόνος έναρξης s_j

Χρονοπρογραμματισμός Διαστημάτων – Άπληστοι Αλγόριθμοι

Άπληστο πρότυπο. Θεωρήστε τις εργασίες σε κάποια σειρά. Επιλέξτε κάθε εργασία εφόσον είναι συμβατή με αυτές που έχουν ήδη επιλεγεί

Δεν αποδίδει ο μικρότερος χρόνος έναρξης s_j

Δεν αποδίδει το μικρότερο χρονικό διάστημα $f_j - s_j$

Χρονοπρογραμματισμός Διαστημάτων – Άπληστοι Αλγόριθμοι

Άπληστο πρότυπο. Θεωρήστε τις εργασίες σε κάποια σειρά. Επιλέξτε κάθε εργασία εφόσον είναι συμβατή με αυτές που έχουν ήδη επιλεγεί

Δεν αποδίδει ο μικρότερος χρόνος έναρξης s_j

Δεν αποδίδει το μικρότερο χρονικό διάστημα $f_j - s_j$

Δεν αποδίδουν οι ελάχιστες διενέξεις

Χρονοπρογραμματισμός Διαστημάτων - Άπληστος Αλγόριθμος

Άπληστος αλγόριθμος. Θεωρήστε τις εργασίες σε αύξουσα σειρά χρόνων λήξης. Επιλέξτε κάθε εργασία εφόσον είναι συμβατή με αυτές που έχουν ήδη επιλεγεί.

```
Ταξινόμησε τις εργασίες κατά χρόνο λήξης έτσι ώστε f1 ≤ f2 ≤ ... ≤ fn

  Επιλεγμένες εργασίες
A ← ∅
for j = 1 to n {
   if (εργασία j συμβατή με το A)
      A ← A ∪ {j}
}
return A
```

Υλοποίηση. O(n log n)

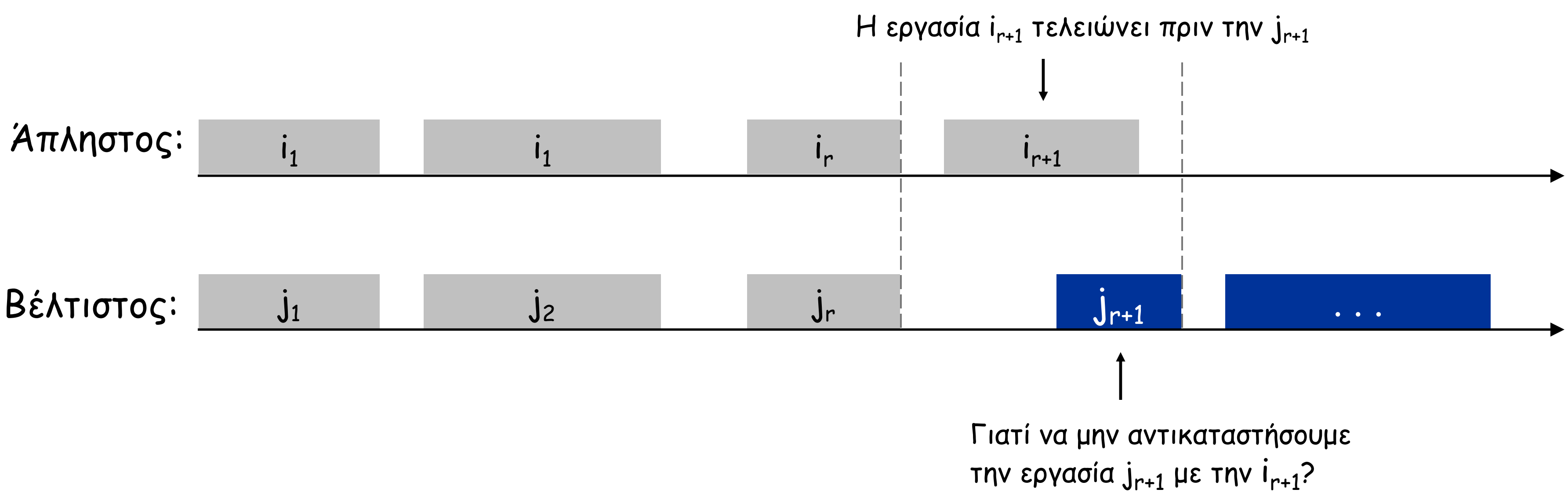
- Θυμηθείτε την εργασία j* που προστέθηκε τελευταία στο *A*
- Η εργασία j είναι συμβατή με το *A* αν $s_j \geq f_{j^*}$

Χρονοπρογραμματισμός Διαστημάτων – Ανάλυση Άπληστου Αλγόριθμου

Θεώρημα. Ο άπληστος αλγόριθμος είναι βέλτιστος.

Απόδειξη. (εις άτοπον απαγωγή)

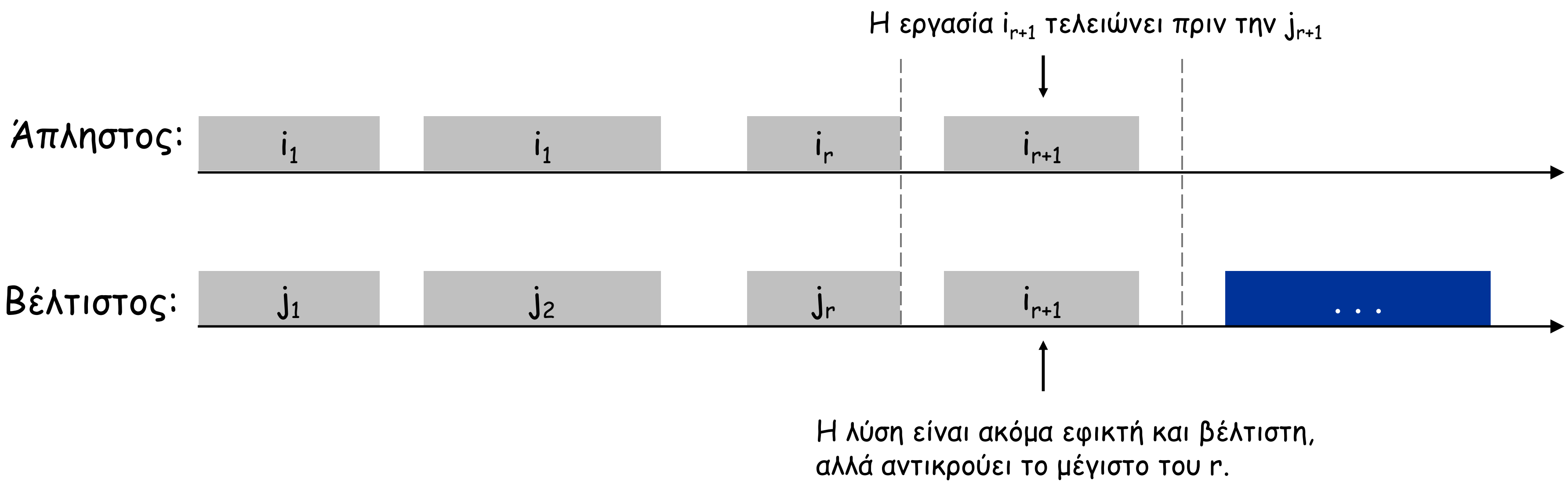
- Έστω ότι ο άπληστος αλγόριθμος δεν είναι βέλτιστος
- Έστω i_1, i_2, ... i_k το σύνολο των εργασιών που επιλέγονται από τον άπληστο αλγόριθμο
- Έστω j_1, j_2, ... j_m το σύνολο των εργασιών στην βέλτιστη λύση με $i_1 = j_1$, $i_2 = j_2$, ..., $i_r = j_r$ για την μέγιστη δυνατή τιμή του r



Θεώρημα. Ο άπληστος αλγόριθμος είναι βέλτιστος.

Απόδειξη. (εις άτοπον απαγωγή)

- Έστω ότι ο άπληστος αλγόριθμος δεν είναι βέλτιστος
- Έστω i_1, i_2, ... i_k το σύνολο των εργασιών που επιλέγονται από τον άπληστο αλγόριθμο
- Έστω j_1, j_2, ... j_m το σύνολο των εργασιών στην βέλτιστη λύση με $i_1 = j_1$, $i_2 = j_2$, ..., $i_r = j_r$ για την μέγιστη δυνατή τιμή του r



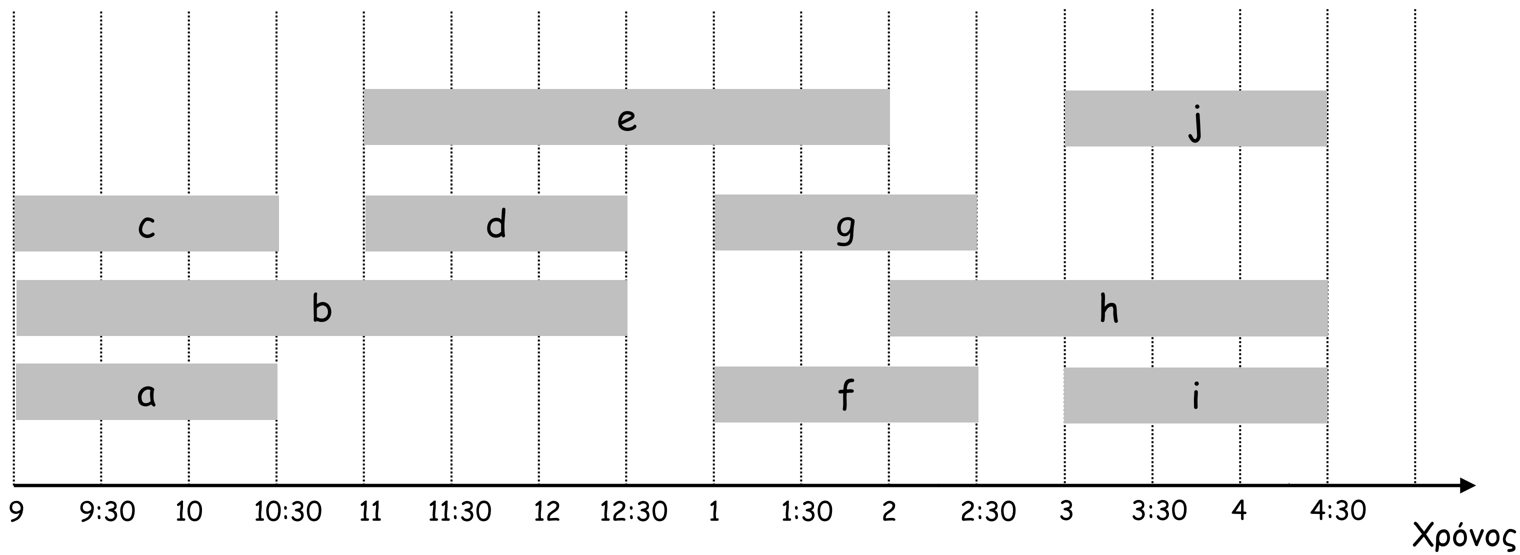
Χρονοπρογραμματισμός όλων των Διαστημάτων
Διαμέριση Χρονικών Διαστημάτων

Διαμέριση Χρονικών Διαστημάτων

Διαμέριση Χρονικών Διαστημάτων.

- Η διάλεξη j ξεκινά τη χρονική στιγμή s_j και τελειώνει τη χρονική στιγμή f_j
- Στόχος: εύρεση ελάχιστου αριθμού αιθουσών για χρονοπρογραμματισμό όλων των διαλέξεων έτσι ώστε να μην υπάρχουν δύο διαλέξεις προγραμματισμένες ταυτόχρονα στην ίδια αίθουσα (συμβατές διαλέξεις).

Π.χ.: Χρονοδιάγραμμα χρήσης 4 αιθουσών για τον προγραμματισμό 10 διαλέξεων

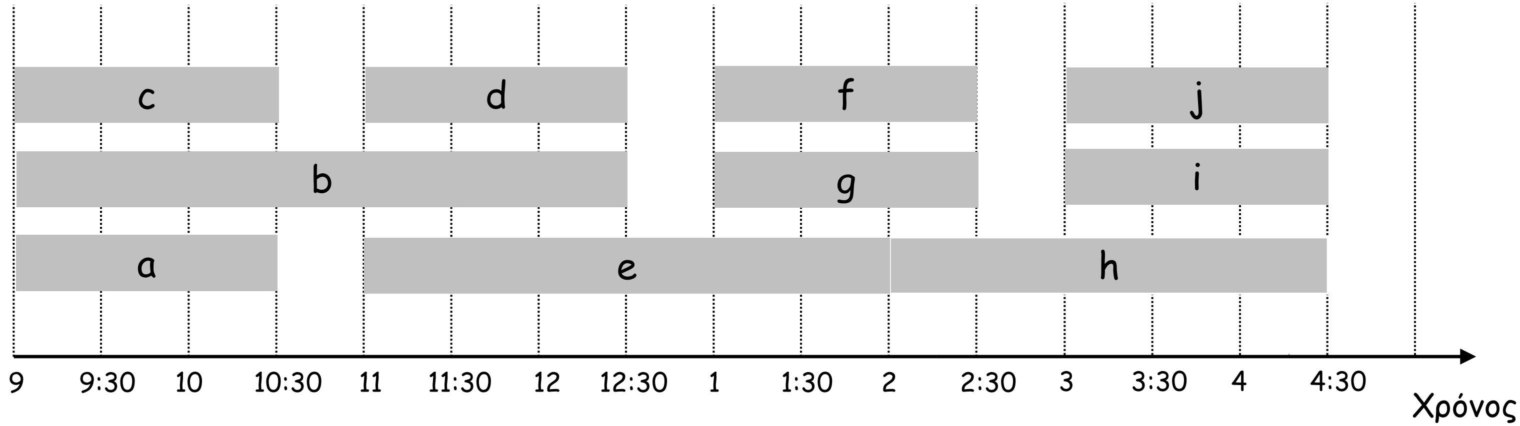


Διαμέριση Χρονικών Διαστημάτων

Διαμέριση Χρονικών Διαστημάτων.

- Η διάλεξη j ξεκινά τη χρονική στιγμή s_j και τελειώνει τη χρονική στιγμή f_j
- Στόχος: εύρεση ελάχιστου αριθμού αιθουσών για χρονοπρογραμματισμό όλων των διαλέξεων έτσι ώστε να μην υπάρχουν δύο διαλέξεις προγραμματισμένες ταυτόχρονα στην ίδια αίθουσα (συμβατές διαλέξεις).

Π.χ.: Αυτό το χρονοδιάγραμμα χρησιμοποιεί μόνο 3 αίθουσες.



Διαμέριση Χρονικών Διαστημάτων – Κάτω όριο στη Βέλτιστη Λύση

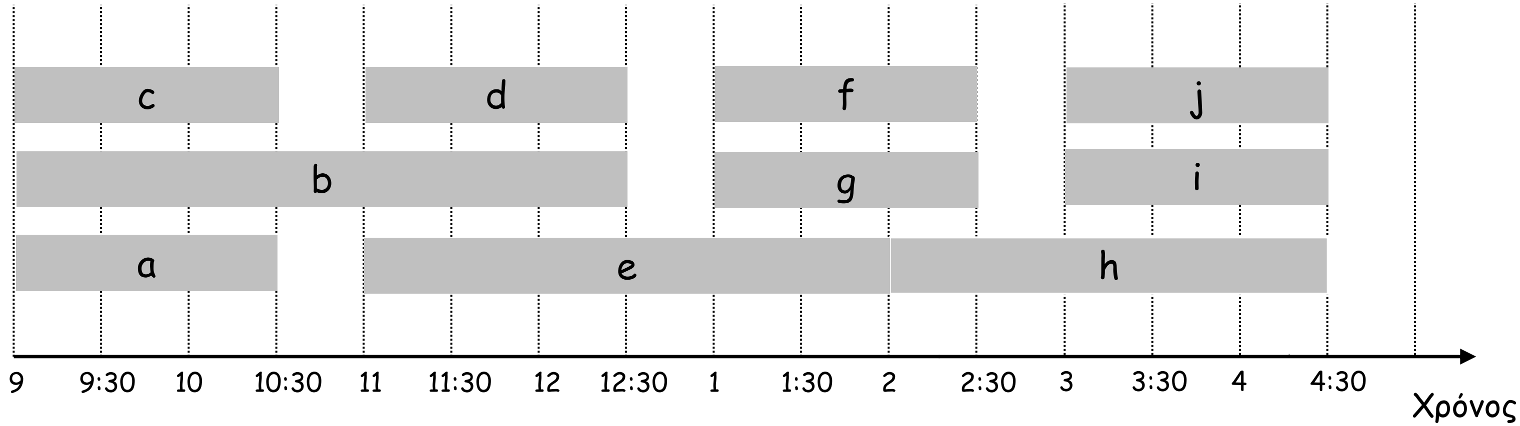
Ορισμός. Το βάθος ενός συνόλου διαστημάτων είναι ο μέγιστος αριθμός διαστημάτων που περιέχουν ένα οποιοδήποτε (κοινό) χρονικό σημείο.

Παρατήρηση. *Ο αριθμός των αιθουσών που χρειάζονται* $\geq$ *βάθος.*

Π.χ.: Βάθος του παρακάτω χρονοδιαγράμματος = 3 $\Rightarrow$ το παρακάτω χρονοδιάγραμμα είναι βέλτιστο.

Τα a, b, c περιέχουν όλα το χρονικό σημείο 9:30

Ερώτηση. Υπάρχει πάντα ένα χρονοδιάγραμμα ίσο με το βάθος των διαστημάτων;



Διαμέριση Χρονικών Διαστημάτων – Άπληστος Αλγόριθμος

Άπληστος Αλγόριθμος. Θεωρήστε τις διαλέξεις σε αύξουσα σειρά χρόνου έναρξης: αναθέστε κάθε διάλεξη σε οποιαδήποτε συμβατή αίθουσα

```
Ταξινόμησε τα διαστήματα κατά χρόνο έναρξης έτσι ώστε s1 ≤
s2 ≤ ... ≤ sn.

d ← 0     ⟵ Αριθμός δεσμευμένων αιθουσών

for j = 1 to n {
   if (διάλεξη j είναι συμβατή με μια αίθουσα k)
      προγραμμάτισε την j στην αίθουσα k
   else
      δέσμευσε μια νέα αίθουσα d + 1
      προγραμμάτισε την j στην αίθουσα d + 1
      d ← d + 1
}
```

Υλοποίηση. $O(n \log n)$

- Για κάθε αίθουσα k, διατηρήστε το χρόνο λήξης της τελευταίας διάλεξης
- Διατηρήστε τις αίθουσες σε μια ουρά προτεραιότητας

Διαμέριση Χρονικών Διαστημάτων – Ανάλυση Άπληστου Αλγόριθμου

Παρατήρηση. *Ο άπληστος αλγόριθμος δεν προγραμματίζει ποτέ δύο ασύμβατες διαλέξεις στην ίδια αίθουσα*

Θεώρημα. *Ο άπληστος αλγόριθμος είναι βέλτιστος.*

Απόδειξη.

- Έστω d = αριθμός αιθουσών που δεσμεύει ο άπληστος αλγόριθμος.
- Η αίθουσα d δεσμεύεται γιατί χρειάζεται να προγραμματίσουμε μια διάλεξη, έστω j, που είναι ασύμβατη με όλες τις υπόλοιπες d-1 αίθουσες.
- Αφού ταξινομήσαμε κατά χρόνο έναρξης, όλες αυτές οι ασυμβατότητες προκαλούνται από διαλέξεις που ξεκινούν όχι αργότερα από τη στιγμή s_j.
- Άρα, υπάρχουν d επικαλυπτόμενες διαλέξεις τη χρονική στιγμή $s_j + \varepsilon$.
- Παρατήρηση $\Rightarrow$ όλα τα προγράμματα χρησιμοποιούν $\geq$ d αίθουσες. ▪

Χρονοπρογραμματισμός για Ελαχιστοποίηση Καθυστέρησης

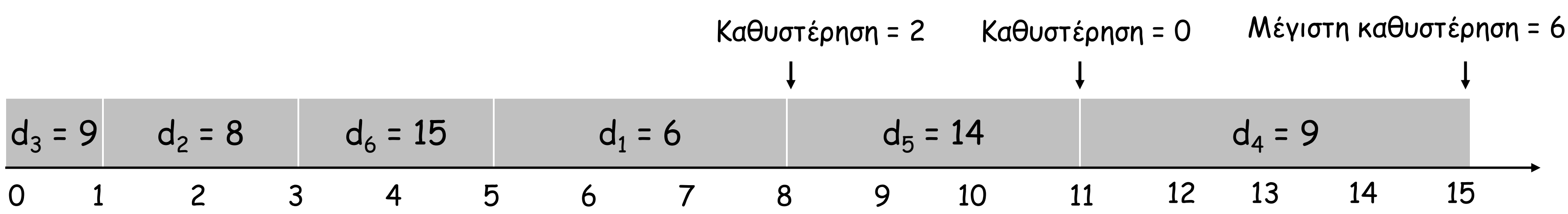
Χρονοπρογραμματισμός για Ελαχιστοποίηση Καθυστέρησης

Πρόβλημα ελαχιστοποίησης καθυστέρησης.

- Ένας υπολογιστικός πόρος επεξεργάζεται μια εργασία την φορά
- Η εργασία j απαιτεί t_j μονάδες χρόνου επεξεργασίας και έχει προθεσμία d_j
- Αν η j ξεκινά την στιγμή s_j, τελειώνει την στιγμή $f_j = s_j + t_j$
- Καθυστέρηση: $\ell_j = \max\{ 0, f_j - d_j \}$
- Στόχος: χρονοπρογραμματισμός όλων των εργασιών για ελαχιστοποίηση της μέγιστης καθυστέρησης $\mathbf{L = \max \ell_j}$

Π.χ.:

	1	2	3	4	5	6
t_j	3	2	1	4	3	2
d_j	6	8	9	9	14	15



Ελαχιστοποίηση Καθυστέρησης – Άπληστοι Αλγόριθμοι

Άπληστο πρότυπο. Θεωρήστε τις εργασίες σε κάποια σειρά.

- [Πρώτα ο μικρότερος χρόνος επεξεργασίας] Θεωρήστε τις εργασίες σε αύξουσα σειρά χρόνου επεξεργασίας t_j

- [Πρώτα η μικρότερη προθεσμία] Θεωρήστε τις εργασίες σε αύξουσα σειρά προθεσμίας d_j

- [Ελάχιστο χρονικό περιθώριο] Θεωρήστε τις εργασίες σε αύξουσα σειρά χρονικού περιθωρίου $d_j - t_j$

Ελαχιστοποίηση Καθυστέρησης – Άπληστοι Αλγόριθμοι

Άπληστο πρότυπο. Θεωρήστε τις εργασίες σε κάποια σειρά.

- [Πρώτα ο μικρότερος χρόνος επεξεργασίας] Θεωρήστε τις εργασίες σε αύξουσα σειρά χρόνου επεξεργασίας t_j

	1	2
t_j	1	10
d_j	100	10

Αντιπαράδειγμα, L=1 (Βέλτιστο L = 0)

- [Ελάχιστο χρονικό περιθώριο] Θεωρήστε τις εργασίες σε αύξουσα σειρά χρονικού περιθωρίου $d_j - t_j$

	1	2
t_j	1	10
d_j	2	10

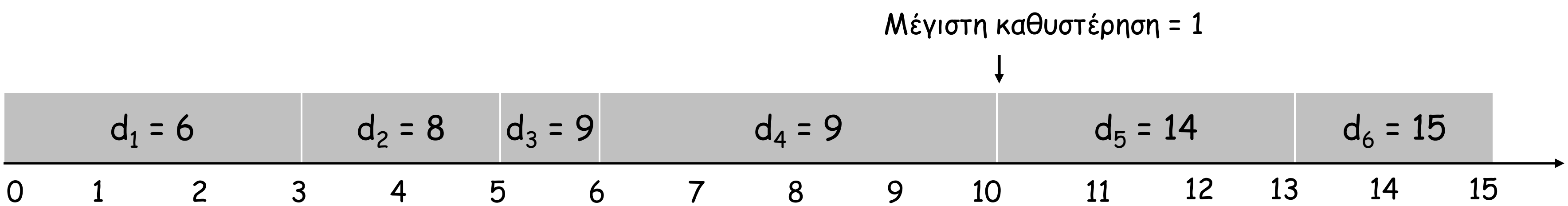
Αντιπαράδειγμα, L = 9 (Βέλτιστο L = 1)

Ελαχιστοποίηση Καθυστέρησης – Άπληστοι αλγόριθμοι

Άπληστος αλγόριθμος. Πρώτα η μικρότερη προθεσμία

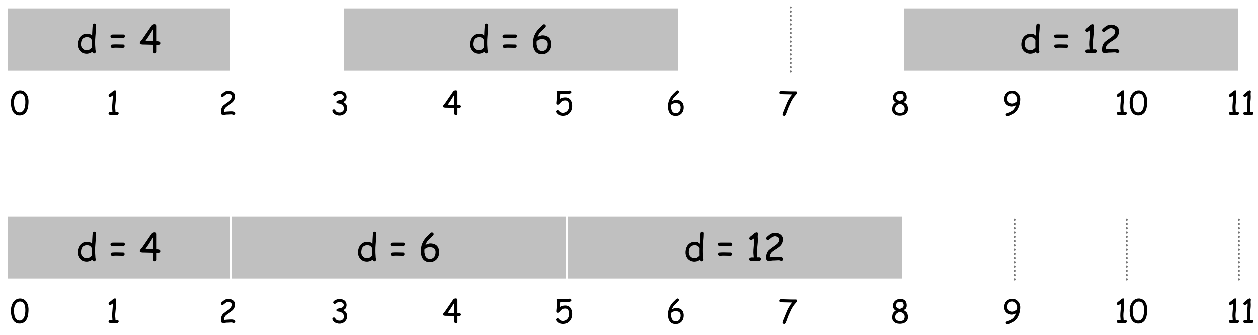
```
Ταξινόμησε n εργασίες κατά προθεσμία έτσι ώστε d1 ≤ d2 ≤ … ≤ dn

t ← 0
for j = 1 to n
   Ανέθεσε την εργασία j στο διάστημα [t, t + tj]
   sj ← t, fj ← t + tj
   t ← t + tj
εκτύπωσε διαστήματα[sj, fj]
```

	1	2	3	4	5	6
t_j	3	2	1	4	3	2
d_j	6	8	9	9	14	15



Ελαχιστοποίηση Καθυστέρησης - Χωρίς χρόνο αδράνειας

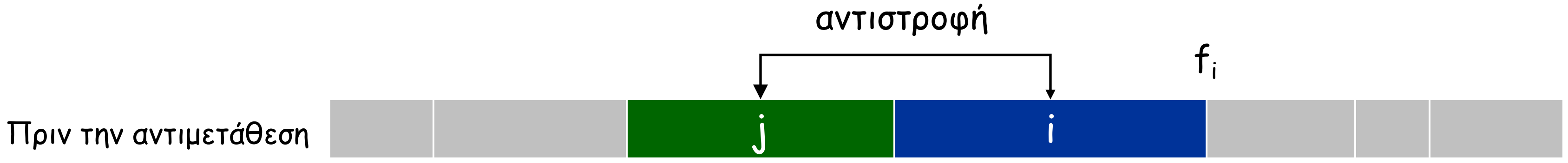
Παρατήρηση. Υπάρχει μια βέλτιστη λύση χωρίς χρόνο αδράνειας



Παρατήρηση. Ο άπληστος αλγόριθμος δεν έχει χρόνο αδράνειας

Ελαχιστοποίηση Καθυστέρησης – Αντιστροφές

Ορισμός. Μια αντιστροφή σε ένα χρονοδιάγραμμα *S* είναι ένα ζευγάρι εργασιών i και j τέτοιο ώστε $d_i < d_j$ αλλά η j χρονοπρογραμματίζεται πριν την i

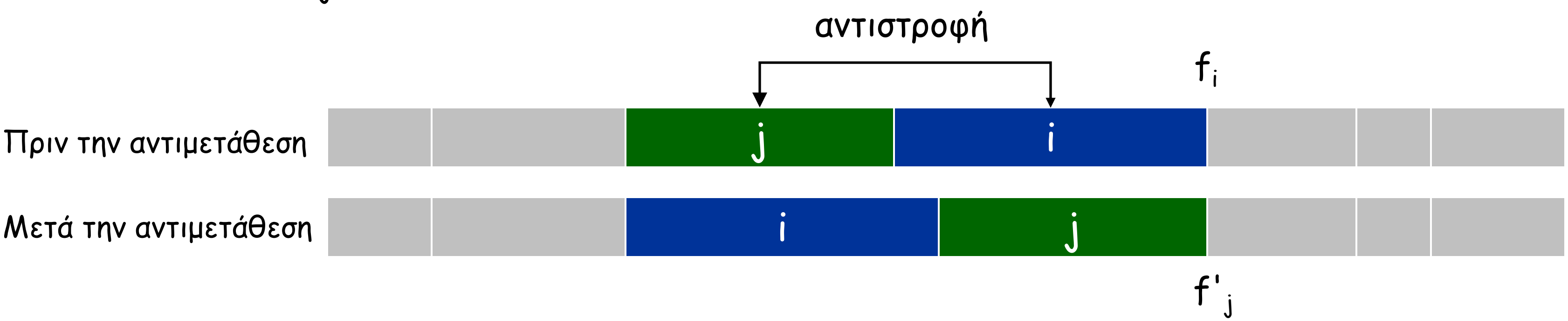


Παρατήρηση. *Ο άπληστος αλγόριθμος δεν έχει αντιστροφές*

Παρατήρηση. *Αν* ένα χρονοδιάγραμμα (χωρίς χρόνο αδράνειας) έχει μια αντιστροφή, τότε έχει μια αντιστροφή με ένα ζευγάρι αντεστραμμένων εργασιών διαδοχικά χρονοπρογραμματισμένων

Ελαχιστοποίηση Καθυστέρησης – Αντιστροφές

Ορισμός. Μια αντιστροφή σε ένα χρονοδιάγραμμα S είναι ένα ζευγάρι εργασιών i και j τέτοιο ώστε $d_i < d_j$ αλλά η j χρονοπρογραμματίζεται πριν την i



Ισχυρισμός. Η αντιμετάθεση δύο γειτονικών, αντεστραμμένων εργασιών μειώνει τον αριθμό των αντιστροφών κατά 1 και δεν αυξάνει την μέγιστη καθυστέρηση

Αποδ. Έστω ℓ η καθυστέρηση πριν την αντιμετάθεση, και έστω ℓ' μετά.

- $\ell'_k = \ell_k$ για κάθε $k \neq i, j$
- $\ell'_i \leq \ell_i$
- Πόσο καθυστερεί η εργασία j:

$$\begin{aligned}
\ell'_j &= f'_j - d_j && \text{(ορισμός)} \\
&= f_i - d_j && \text{(η j τελειώνει τη στιγμή } f_i\text{)} \\
&\leq f_i - d_i && (i < j) \\
&\leq \ell_i && \text{(ορισμός)}
\end{aligned}$$

Ελαχιστοποίηση Καθυστέρησης – Ανάλυση Άπληστου Αλγόριθμου

Θεώρημα. Το χρονοδιάγραμμα S του άπληστου αλγόριθμου είναι βέλτιστο

Απόδειξη. Ορίζουμε το S^* να είναι ένα βέλτιστο χρονοδιάγραμμα που έχει τον ελάχιστο αριθμό αντιστροφών.

- Υποθέτουμε ότι το S^* δεν έχει χρόνο αδράνειας.
- Αν το S^* δεν έχει αντιστροφές, τότε $S = S^*$.
- Αν το S^* έχει μια αντιστροφή, τότε έστω i-j μια γειτονική αντιστροφή.
  - Αντιμεταθέτοντας την i και j δεν αυξάνει την μέγιστη καθυστέρηση και μειώνει αυστηρά τον αριθμό των αντιστροφών
  - Αυτό έρχεται σε αντίφαση με τον ορισμό του S^* ▪

Στρατηγικές Ανάλυσης Άπληστων Αλγορίθμων

Ο άπληστος αλγόριθμος υπερτερεί. Δείξτε ότι μετά από κάθε βήμα του άπληστου αλγόριθμου, η λύση του είναι τουλάχιστον τόσο καλή όσο οποιουδήποτε άλλου αλγόριθμου.

Επιχείρημα ανταλλαγής. Μετατρέψτε σταδιακά οποιαδήποτε λύση σε αυτή που ανακαλύπτεται από τον άπληστο αλγόριθμο χωρίς να επηρεάζεται η ποιότητά της.

Δομική. Ανακαλύψτε ένα απλό "δομικό" όριο που να επιβεβαιώνει ότι κάθε πιθανή λύση πρέπει να έχει μια συγκεκριμένη τιμή. Στη συνέχεια, δείξτε ότι ο αλγόριθμος σας επιτυγχάνει πάντα αυτό το όριο.

Βιβλιογραφία

1. J. Kleinberg and E. Tardos, *Σχεδιασμός Αλγορίθμων*, ελληνική έκδοση, Εκδόσεις Κλειδάριθμος, 2008
2. T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Εισαγωγή στους Αλγορίθμους*, ελληνική έκδοση, Πανεπιστημιακές Εκδόσεις Κρήτης, 2012
3. K. Mehlhorn and P. Sanders, *Αλγόριθμοι και Δομές Δεδομένων – Τα βασικά εργαλεία*, ελληνική έκδοση, Εκδόσεις Κλειδάριθμος, 2014
4. S. Dasgupta, C. Papadimitriou, and U. Vazirani, *Αλγόριθμοι*, ελληνική έκδοση, Εκδόσεις Κλειδάριθμος, 2008
5. Θ. Παπαθεοδώρου, *Αλγόριθμοι: Εισαγωγικά Θέματα και Παραδείγματα*, Εκδόσεις Πανεπιστημίου Πατρών, 1999

Τέλος Ενότητας

ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Σημείωμα Ιστορικού Εκδόσεων Έργου

Το παρόν έργο αποτελεί την έκδοση 1.0.

Σημείωμα Αναφοράς

«Εισαγωγή στους Αλγορίθμους». Έκδοση: 1.0. Πάτρα 2014.
Διαθέσιμο από τη δικτυακή διεύθυνση:

https://eclass.upatras.gr/courses/CEID1083

Σημείωμα Αδειοδότησης

Πηγές εικόνων - Χρήση Έργων Τρίτων

Εικόνα 1: σελ. 6,

http://en.wikipedia.org/wiki/File:Wall_Street_film.jpg

Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.