ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ

Αλγόριθμοι και πολυπλοκότητα

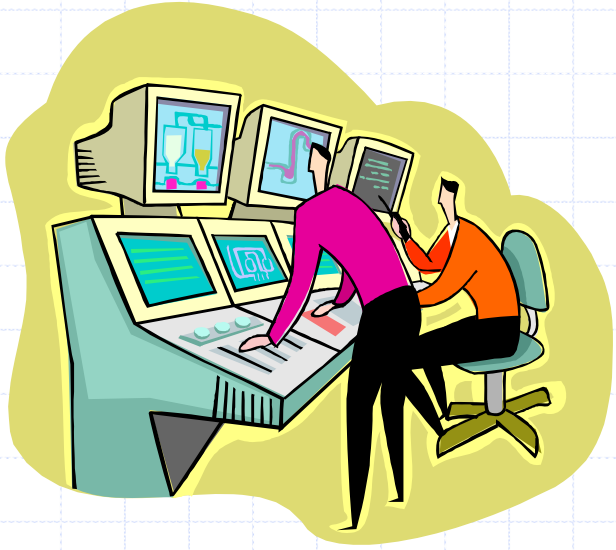
Δυναμικός Προγραμματισμός

Ιωάννης Τόλλης
Τμήμα Επιστήμης Υπολογιστών

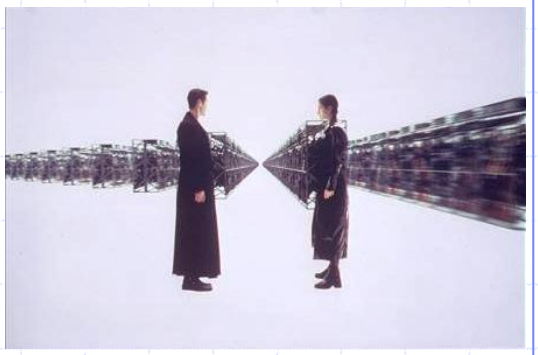
Δυναμικός Προγραμματισμός

Περίληψη και ανάγνωση

- Πολλαπλασιασμός αλληλουχίας πινάκων
- Η γενική τεχνική
- Το ακέραιο πρόβλημα του σακιδίου

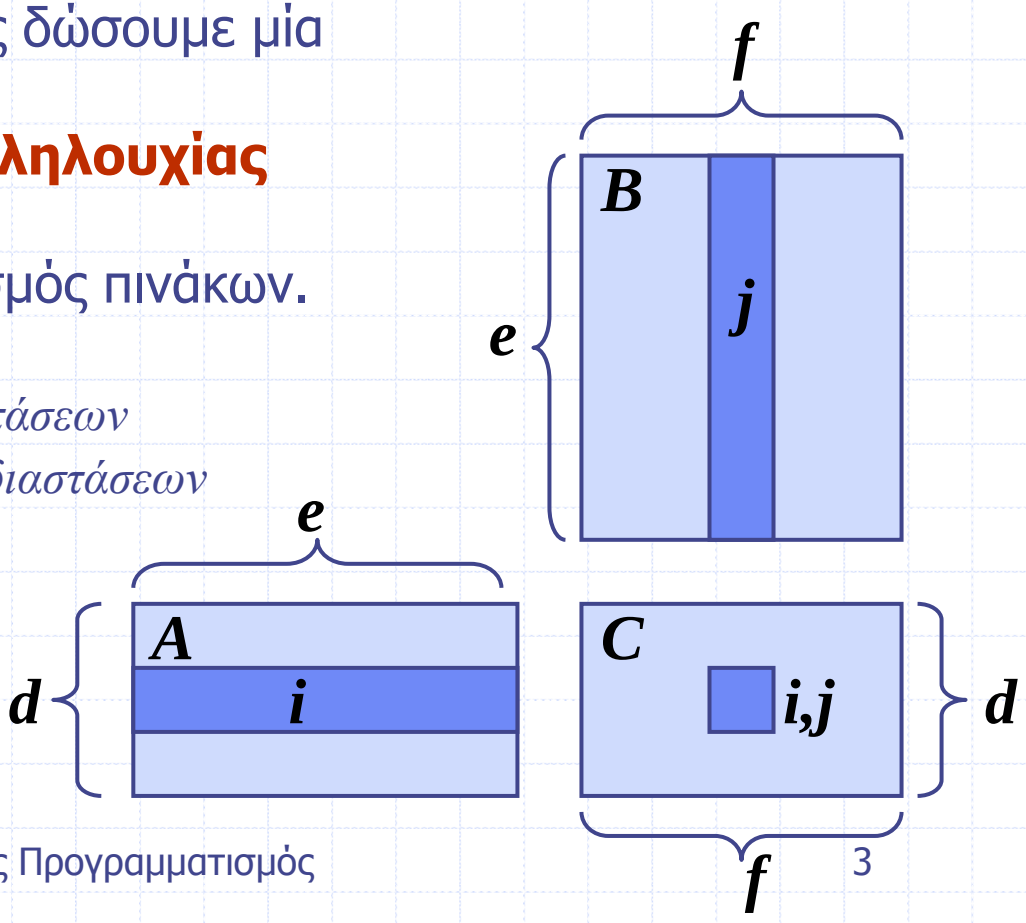


Πολλαπλασιασμός αλληλουχίας πινάκων



- Δυναμικός προγραμματισμός είναι ένα γενικό παράδειγμα σχεδίασης αλγορίθμου.
  - Είναι καλύτερα να σας δώσουμε ένα παράδειγμα, αντί να σας δώσουμε μία γενική τεχνική:
  - **Πολλαπλασιασμός αλληλουχίας πινάκων**
- Ανασκόπηση:Πολλαπλασιασμός πινάκων.
  - $C = A*B$
  - *Ο πίνακας* A *είναι* $d \times e$ *διαστάσεων και ο πίνακας* B *είναι* $e \times f$ *διαστάσεων*
  - *Ο χρόνος είναι* $O(d \cdot e \cdot f)$

$$C[i,j] = \sum_{k=0}^{e-1} A[i,k] * B[k,j]$$



Πολλαπλασιασμός αλληλουχίας πινάκων

- **Πολλαπλασιασμός αλληλουχίας πινάκων:**
  - Υπολογίζουμε το γινόμενο $A=A_0*A_1*...*A_{n-1}$
  - Οι πίνακες A_i είναι διαστάσεων $d_i \times d_{i+1}$
  - Πρόβλημα:Πώς ομαδοποιούμε παρενθετικά τους όρους;
- Παράδειγμα
  - Ο πίνακας B είναι διαστάσεων 3×100
  - Ο πίνακας C είναι διαστάσεων 100×5
  - Ο πίνακας D είναι διαστάσεων 5×5
  - Το γινόμενο (B*C)*D εκτελεί 1500 + 75 = 1575 διαδικασίες
  - Το γινόμενο B*(C*D) εκτελεί 1500 + 2500 = 4000 διαδικασίες

Προσέγγιση απαρίθμησης

- **Αλγόριθμος του πολ/μου αλληλουχίας πινάκων**
  - Δοκιμάστε όλους τους πιθανούς τρόπους να ομαδοποιήσετε παρενθετικά το $A=A_0*A_1*...*A_{n-1}$
  - Υπολογίστε των αριθμό των διαδικασιών για κάθε ένα από τους πιθανούς τρόπους.
  - Επιλέξτε εκείνον που είναι καλύτερος
- Χρόνος Εκτέλεσης:
  - Ο αριθμός των ομαδοποιήσεων είναι ίσος με το αριθμό των δυαδικών δέντρων με n κόμβους
  - Αυτό είναι εκθετικό!
  - Καλείται αριθμός Catalan, είναι σχεδόν 4^n.
  - Αυτός δεν είναι καθόλου ικανοποιητικός αλγόριθμος!

Η άπληστη προσέγγιση

- Ιδέα #1: επιλέξτε επαναληπτικά το προϊόν που χρησιμοποιεί τις περισσότερες διαδικασίες.
- Αντιπαράδειγμα:
  - Ο πίνακας A είναι 10 × 5
  - Ο πίνακας B είναι 5 × 10
  - Ο πίνακας C είναι 10 × 5
  - Ο πίνακας D είναι 5 × 10
  - Άπληστη Ιδέα #1 δίνει (A*B)*(C*D), το οποίο εκτελεί 500+1000+500 = 2000 διαδικασίες
  - A*((B*C)*D) εκτελεί 500+250+250 = 1000 διαδικασίες

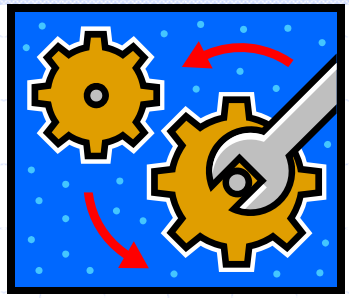
Άλλη μία άπληστη προσέγγιση

- Ιδέα #2: επιλέξτε επαναληπτικά το προϊόν που χρησιμοποιεί τις λιγότερες διαδικασίες.
- Αντιπαράδειγμα:
  - Ο πίνακας A είναι 101 × 11
  - Ο πίνακας B είναι 11 × 9
  - Ο πίνακας C είναι 9 × 100
  - Ο πίνακας D είναι 100 × 99
  - Άπληστη ιδέα #2 δίνει A*((B*C)*D)), το οποίο εκτελεί 109989+9900+108900=228789 διαδικασίες
  - (A*B)*(C*D) εκτελεί 9999+89991+89100=189090 διαδικασίες
- Η άπληστη προσέγγιση δεν μας δίνει την βέλτιστη τιμή.

Αναδρομική Προσέγγιση

- Καθορισμός **υπο-προβλημάτων**:
  - Βρείτε τον καλύτερο τρόπο παρενθετικής ομαδοποίησης του γινομένου $A_i*A_{i+1}*...*A_j$.
  - Έστω $N_{i,j}$ το πλήθος των διαδικασιών που έγιναν από αυτό το υπο-πρόβλημα.
  - Η βέλτιστη λύση για όλο το πρόβλημα είναι $N_{0,n-1}$.
- **Βελτιστοποίηση υπο-προβλήματος**: Η βέλτιστη λύση μπορεί να καθοριστεί από τα βέλτιστα υπο-προβλήματα.
  - Πρέπει να υπάρξει ένας τελικός πολλαπλασιασμός(ρίζα του δέντρου έκφρασης).
  - Ο τελικός πολλαπλασιασμός είναι στον δείκτη i : $(A_0*...*A_i)*(A_{i+1}*...*A_{n-1})$.
  - Τότε η βέλτιστη λύση $N_{0,n-1}$ είναι το άθροισμα των δύο βέλτιστων υπο-προβλημάτων, $N_{0,i}$ και $N_{i+1,n-1}$ συν το χρόνο τον τελευταίο πολλαπλασιασμό.
  - Αν η global βέλτιστη δεν είχε αυτά τα βέλτιστα υπο-προβλήματα, θα μπορούσαμε να καθορίσουμε μία ακόμα καλύτερη «βέλτιστη» λύση.

Χαρακτηρισμός της εξίσωσης

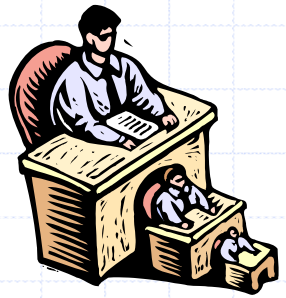


- Το ολικό βέλτιστο πρέπει να καθοριστεί από τα βέλτιστα υπο-προβλήματα, ανάλογα με το που είναι ο τελικός πολλαπλασιασμός.
- Εξετάζουμε όλες τις πιθανές θέσεις για τον τελικό προγραμματισμό:
  - Θυμηθείτε ότι ο A_i είναι ένας $d_i \times d_{i+1}$ διαστάσεων πίνακας.
  - Έτσι, ένας χαρακτηρισμός εξίσωσης για το $N_{i,j}$ είναι ο παρακάτω:

$$N_{i,j} = \min_{i \le k < j} \{ N_{i,k} + N_{k+1,j} + d_i d_{k+1} d_{j+1} \}$$

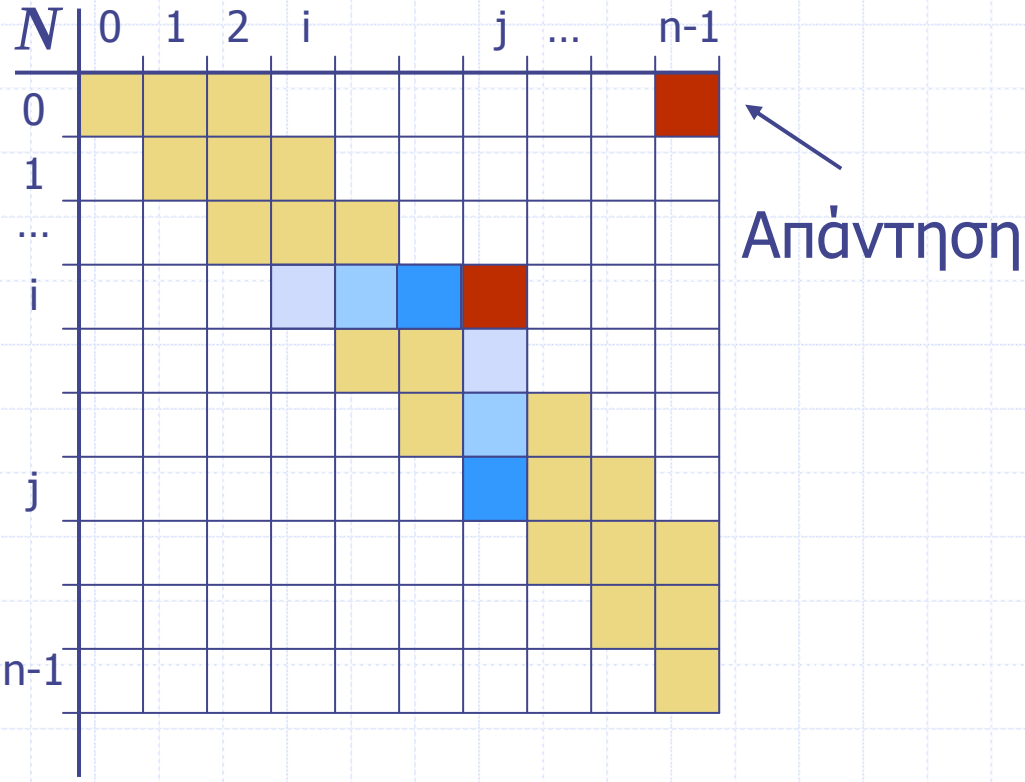
- Σημειώστε ότι τα υπο-προβλήματα δεν είναι ανεξάρτητα – **Επικάλυψη υπο-προβλημάτων**.

Δυναμικός Προγραμματισμός Απεικόνιση Αλγορίθμου

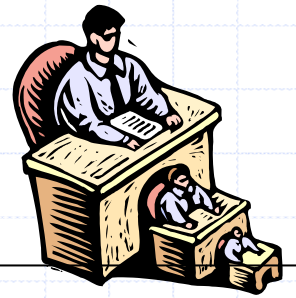


- Η από κάτω προς τα πάνω κατασκευή συμπληρώνει τους Ν πίνακες από τις διαγώνιες
- Το $N_{i,j}$ παίρνει τιμές από τις προηγούμενες καταχωρήσεις στην i-οστη γραμμή και στην j-οστη στήλη.
- Συμπληρώνοντας σε κάθε καταχώρηση στον Ν πίνακα παίρνει Ο(n) χρόνο.
- Συνολικός χρόνος εκτέλεσης: $O(n^3)$
- Το να πάρουμε πραγματική παρενθετική ομαδοποίηση μπορεί να γίνει με το "k" για κάθε Ν καταχώρηση

$$N_{i,j} = \min_{i \le k < j} \{ N_{i,k} + N_{k+1,j} + d_i d_{k+1} d_{j+1} \}$$



Δυναμικός Προγραμματισμός Αλγόριθμος



- Με την επικάλυψη υπο-προβλημάτων, δεν χρησιμοποιούμε αναδρομή.
- Αντ' αυτού, κατασκευάζουμε βέλτιστα υπο-προβλήματα "από κάτω προ τα πάνω."
- Των $N_{i,I}$ είναι εύκολα, όποτε άς ξεκινήσουμε με αυτά
- Έπειτα κάντε τα προβλήματα μήκους 2,3,... subproblems, και συνεχίστε.
- Χρόνος εκτέλεσης: $O(n^3)$

Algorithm ***matrixChain***(S):

Input: ακολουθία S n πινάκων που θα πολλαπλασιαστούν.

Output: αριθμός διαδικασιών σε μία βέλτιστη παρενθετική ομαδοποίηση της S

for $i \leftarrow 1$ **to** $n-1$ **do**

  $N_{i,i} \leftarrow 0$

for $b \leftarrow 1$ **to** $n-1$ **do**

  { $b = j - i$ είναι το μήκος του προβλήματος }

  for $i \leftarrow 0$ **to** $n-b-1$ **do**

    $j \leftarrow i + b$

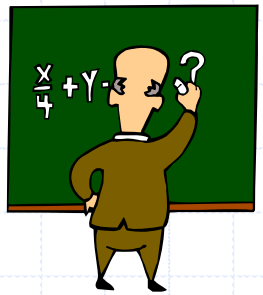
    $N_{i,j} \leftarrow +\infty$

    for $k \leftarrow i$ **to** $j-1$ **do**

      $N_{i,j} \leftarrow \min\{N_{i,j}, N_{i,k} + N_{k+1,j} + d_i d_{k+1} d_{j+1}\}$

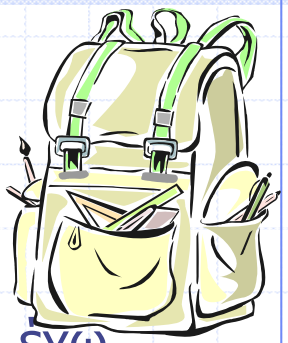
return $N_{0,n-1}$

Η Γενική Τεχνική Δυναμικού Προγραμματισμού



- Ισχύει για ένα πρόβλημα που αρχικά φαίνεται να απαιτεί πολύ χρόνο (πιθανόν εκθετικό), υπό τον όρο να έχουμε:
  - **Απλά υπο-προβλήματα:** τα υπο-προβλήματα μπορούν να οριστούν από λίγες μεταβλητές, όπως j, k, l, m κ.τ.λ.
  - **Βελτιστοποίηση υπο-προβλημάτων :** η global βέλτιστη τιμή μπορεί να οριστεί από βέλτιστα υπο-προβλήματα
  - **Επικάλυψη υπο-προβλημάτων :** τα υπο-προβλήματα δεν είναι ανεξάρτητα, αλλά επικαλυπτόμενα (ως εκ τούτου,πρέπει να κατασκευαστούν από κάτω προς τα πάνω).

Το ακέραιο πρόβλημα του σακιδίου

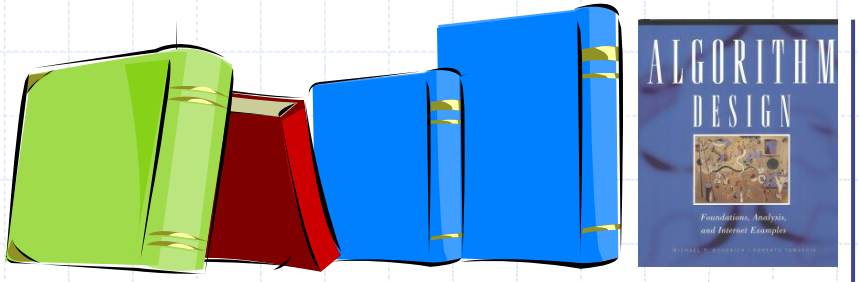


- Δεδομένα: ένα σύνολο S από n στοιχεία, με κάθε στοιχείο έχω
  - w_i – ένα θετικό βάρος
  - b_i – ένα θετικό όφελος
- Σκοπός: διαλέξτε στοιχεία με μέγιστο συνολικό όφελος αλλά με βάρος το πολύ W.
- Αν **δεν** μας επιτρέπεται να παίρνουμε κλασματικά ποσά, τότε αυτό είναι το **ακέραιο πρόβλημα του σακιδίου.**
  - Σε αυτή την περίπτωση, το T δηλώνει το σύνολο των στοιχείων που πήραμε
  - Στόχος: μεγιστοποίηση του $\sum_{i \in T} b_i$
  - Περιορισμός: $\sum_{i \in T} w_i \leq W$

Παράδειγμα

- Δεδομένα: Ένα σύνολο S από n στοιχεία, με κάθε στοιχείο έχω
  - b_i – ένα θετικό "όφελος"
  - w_i - ένα θετικό "βάρος"
- Σκοπός: διαλέξτε στοιχεία με μέγιστο συνολικό όφελος αλλά με βάρος το πολύ W.

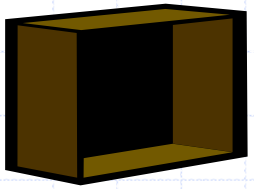
Στοιχεία:



1 2 3 4 5

Βάρη:	4 in	2 in	2 in	6 in	2 in
Οφέλη:	$20	$3	$6	$25	$80

"σακίδιο"



κουτί βάρους 9

Λύση:

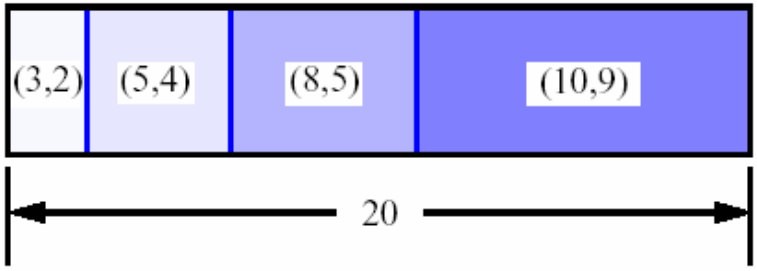
- στοιχείο 5 ($80, 2 in)
- στοιχείο 3 ($6, 2in)
- στοιχείο 1 ($20, 4in)

Ένας αλγόριθμος του ακέραιου προβλήματος του σακιδίου, Πρώτη Προσπάθεια

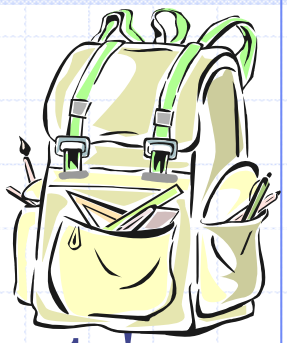
- S_k: Σύνολό από στοιχεία που αριθμούνται από το 1 έως το k.
- Καθορισμός του B[k] = καλύτερη επιλογή από το S_k.
- Πρόβλημα: δεν έχει βελτιστοποίηση υπο-προβλημάτων:
  - Εξετάζουμε το σύνολο S={(3,2),(5,4),(8,5),(4,3),(10,9)} από (όφελος, βάρος) ζευγάρια και συνολικό βάρος W = 20

Καλύτερο για S_4:

Καλύτερο για S_5:



Ένας αλγόριθμος του ακέραιου προβλήματος του σακιδίου, Δεύτερη Προσπάθεια



- S_k: Σύνολό από στοιχεία που αριθμούνται από το 1 έως το k.
- Καθορισμός του B[k,w] ως η καλύτερη επιλογή από το S_k με βάρος το πολύ w
- Καλά νέα: Αυτό έχει βελτιστοποίηση υπο-προβλήματων.

$$B[k,w] = \begin{cases} B[k-1,w] & \text{if } w_k > w \\ \max\{B[k-1,w],\ B[k-1,w-w_k]+b_k\} & \text{else} \end{cases}$$

- Π.χ., το καλύτερο υποσύνολο του S_k με βάρος το πολύ w είναι ή
  - το καλύτερο υποσύνολο του S_{k-1} με βάρος το πολύ w ή
  - το καλύτερο υποσύνολο του S_{k-1} με βάρος το πολύ $w–w_k$ συν το στοιχείο k

Αλγόριθμος του ακέραιου προβλήματος του σακιδίου

$$B[k,w] = \begin{cases} B[k-1,w] & \text{if } w_k > w \\ \max\{B[k-1,w],\ B[k-1,w-w_k]+b_k\} & \text{else} \end{cases}$$

- Θυμηθείτε των καθορισμό του B[k,w]
- Όταν το B[k,w] καθοριστεί από τα B[k–1,*], μπορούμε να χρησιμοποιήσουμε δύο πίνακες αντί για έναν
- Χρόνος εκτέλεσης: O(nW).
- Δεν είναι αλγόριθμος πολυωνυμικού χρόνου ενώ το W μπορεί να είναι μεγαλύτερο
- Αυτό είναι ένας αλγόριθμος ψεύδο-πολυωνυμικού χρόνου

Algorithm ***01Knapsack(S, W)***:

Input: σύνολο S από n στοιχεία με όφελος b_i και βάρος w_i; Μέγιστο βάρος W

Output: όφελος του καλύτερου υποσυνόλου του S με βάρος το πολύ W

let A and B be arrays of length $W + 1$

for $w \leftarrow 0$ **to** W **do**

$B[w] \leftarrow 0$

for $k \leftarrow 1$ **to** n **do**

αντιγραφή του πίνακα B στον πίνακα A

for $w \leftarrow w_k$ **to** W **do**

if $A[w–w_k] + b_k > A[w]$ **then**

$B[w] \leftarrow A[w–w_k] + b_k$

return $B[W]$

Τέλος Ενότητας

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Κρήτης» 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 ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Σημειώματα

Σημείωμα αδειοδότησης

- Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση, Όχι Παράγωγο Έργο 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».

[1] http://creativecommons.org/licenses/by-nc-nd/4.0/

- Ως Μη Εμπορική ορίζεται η χρήση:
  - που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
  - που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
  - που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

- Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

Σημείωμα Αναφοράς

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.