ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

ΑΝΟΙΚΤΑ
ΑΚΑΔΗΜΑΪΚΑ
ΜΑΘΗΜΑΤΑ

ΑΛΓΟΡΙΘΜΟΙ

Ενότητα 3: Ωμή Βία

Ιωάννης Μανωλόπουλος, Καθηγητής
Αναστάσιος Γούναρης, Επίκουρος Καθηγητής
Τμήμα Πληροφορικής ΑΠΘ

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

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης» 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

Ωμή Βία

Προβλήματα, αλγόριθμοι και αποδοτικότητα

BY NC SA
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

Ωμή Βία

Είναι μία άμεση προσέγγιση που βασίζεται στην εκφώνηση του προβλήματος και τους ορισμούς των εννοιών

Παραδείγματα:

1. Υπολογισμός του a^n (όπου a, $n > 0$, ακέραιοι)
2. ΜΚΔ με διαδοχικό έλεγχο ακεραίων
3. Υπολογισμός του n!
4. Πολλαπλασιασμός δύο πινάκων n X n
5. Ταξινόμηση με επιλογή
6. Σειριακή αναζήτηση

Σειριακή αναζήτηση

Algorithm Sequential(A[0..n-1])

```
// Input: An array A[0..n-1] and a sought value k
// Output: the position of k if found, -1 otherwise
i ← 0
while i<n and A[i]<>k do
   i ← i+1
If i<n return i
else return -1
```

- Χειρότερη περίπτωση
- Καλύτερη περίπτωση
- Μέση περίπτωση

Bubble Sort

Algorithm BubbleSort(A[0..n-1])

```
// Input: an array A[0..n-1] of orderable elements
// Output: Array A[0..n-1] sorted ascendingly
for i ← 0 to n-2 do
   for j ← 0 to n-2-i do
      if A[j+1]<A[j] swap A[j] and A[j+1]
```

- Ανάλυση για συγκρίσεις και μετακινήσεις
- Βελτιώσεις

Ταύτιση Προτύπου

- Πρότυπο: ένα προς αναζήτηση string από *m* χαρακτήρες
- Κείμενο: ένα μεγάλο string από *n* χαρακτήρες, όπου θα γίνει η αναζήτηση
- Αλγόριθμος ωμής βίας:

1. Ευθυγραμμίζουμε το πρότυπο με την αρχή του κειμένου
2. Μετακινώντας από αριστερά προς τα δεξιά, συγκρίνουμε κάθε χαρακτήρα του προτύπου με τον αντίστοιχο χαρακτήρα του κειμένου μέχρι
    - Να ταυτίζονται όλοι οι χαρακτήρες (επιτυχής αναζήτηση); ή
    - Να διαπιστωθεί μία ασυμφωνία
3. Ενόσω το πρότυπο δεν βρίσκεται και το κείμενο δεν έχει εξαντληθεί, επανα-ευθυγραμμίζουμε το πρότυπο κατά μία θέση προς τα δεξιά και πηγαίνουμε στο Βήμα 2.

Αλγόριθμος Ωμής Βίας για Ταύτιση Προτύπου

<u>Algorithm BruteForceStringMatch(T[0..n-1], P[0..m-1])</u>

```
// Input: array T[0..n-1] for text and array P[0..m-
   1] for pattern

// Output: the position of the first character in
   the text stat starts the first matching substring
   if the search is successful, and -1 otherwise

for i ← 0 to n-m do

   j ← 0

   while j<m and P[j]=T[i+j] do

      j ← j+1

   if j=m return i

return -1
```

Αλγόριθμος Ωμής Βίας για Ταύτιση Προτύπου (2)

Παραδείγματα

- Pattern: 001011
  Text: 10010101101001100101111010
- Pattern: happy
  Text: It is never too late to have a happy childhood

Αριθμός συγκρίσεων ;

Αποδοτικότητα ;

Αλγόριθμος Ωμής Βίας για αποτίμηση πολυωνύμου

- Πρόβλημα: Να βρεθεί η τιμή του πολυωνύμου $p(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x^1 + a_0$

  στο σημείο $x = x_0$
- <u>Αλγόριθμος</u>

```
p := 0.0
for i := n down to 0 do
    power := 1
    for  j := 1 to i do
          power :=  power * x
    p := p + a[i] * power
return p
```

- <u>Αποδοτικότητα</u> ;

Αποτίμηση πολυωνύμου – βελτίωση

- Βελτίωση: υπολογισμός από αριστερά προς τα δεξιά
- Αλγόριθμος:

```
p := a[0]
power := 1

for i := 1 to n do
   power := power * x
   p := p + a[i] * power
return p
```

- Αποδοτικότητα ;

Το πρόβλημα του πλησιέστερου ζεύγους

- Πρόβλημα: να βρεθούν τα δύο πλησιέστερα σημεία μεταξύ *n* σημείων σε ένα *k*-διάστατο χώρο
- Αλγόριθμος:

```
// Input: List P of n>1 points
   P1=(x1,y1),…,Pn=(xn,yn)
// Output: Indices id1, id2 of the closest pair
dmin ← ∞
for i ← 1 to n-1 do
    for j← i+1 to n do
          d ← sqrt[(xi-xj)²+(yi-yj)²]
          if d<dmin
                dmin←d; id1←i; id2←j;
return id1, id2
```

- Αποδοτικότητα (τετραγωνική ρίζα) ;

Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

Το πρόβλημα του κυρτού περιβλήματος

- <u>Πρόβλημα</u>: να βρεθεί το μικρότερο κυρτό πολύγωνο που περικλείει *n* σημεία στο επίπεδο
- <u>Αλγόριθμος</u>: Για κάθε ζεύγος σημείων p_1 και p_2 προσδιορίζουμε αν όλα τα άλλα σημεία ανήκουν στο ίδιο ημιεπίπεδο σε σχέση με την ευθεία που διέρχεται από τα p_1 και p_2
- <u>Μέθοδος:</u> Τα σημεία (x_1,y_1) και (x_2,y_2) ορίζουν μία ευθεία $ax+by=c$, όπου $a=y_2-y_1$, $b=x_1-x_2$ και $c=x_1y_2-y_1x_2$
- <u>Αποδοτικότητα</u> ;

Δυνατά και αδύνατα σημεία ωμής βίας

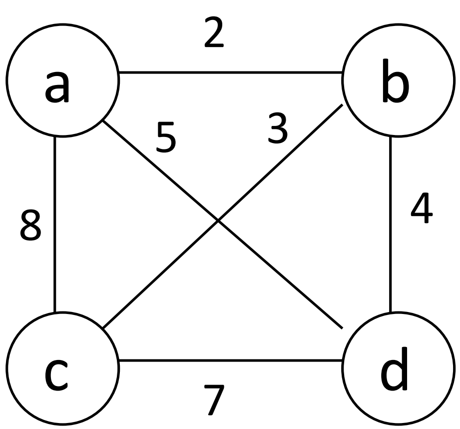
- Δυνατά:
  - Μεγάλη εφαρμοσιμότητα και απλότητα
  - Δίνει λογικούς αλγορίθμους για σημαντικά προβλήματα
    - αναζήτηση, ταύτιση προτύπου, πολλαπλασιασμός πινάκων
  - Δίνει πρότυπους αλγορίθμους για απλά προβλήματα
    - Άθροισμα/γινόμενο n αριθμών, εύρεση μέγιστου/ελάχιστου σε λίστα
- Αδυναμίες:
  - Δίνει σπάνια αποτελεσματικούς αλγορίθμους
  - Μερικοί αλγόριθμοι ωμής βίας είναι απαράδεκτα αργοί
  - Δεν είναι τόσο δημιουργική όσο άλλες σχεδιαστικές τεχνικές

Εξαντλητική αναζήτηση

- Εξαντλητική αναζήτηση είναι μία μέθοδος ωμής βίας όπου αναζητούμε ένα στοιχείο με συγκεκριμένη ιδιότητα, συχνά μεταξύ συνδυαστικών αντικειμένων όπως συνδυασμούς, διατάξεις ή υποσύνολα συνόλου
- Μέθοδος:
  – Κατασκευάζουμε μία λίστα όλων των δυνατών λύσεων στο πρόβλημα με ένα συστηματικό τρόπο
    - Παρουσιάζονται όλες οι λύσεις
    - Καμία λύση δεν επαναλαμβάνεται
  – Εξετάζουμε τις λύσεις μία προς μία, απορρίπτοντας τις μη ικανοποιητικές και κρατώντας τη μέχρι στιγμής καλύτερη
  – Όταν τελειώσει η αναζήτηση, ανακοινώνεται ο νικητής

Παράδειγμα 1: πρόβλημα περιοδεύοντος πωλητού

- Πρόβλημα: Δεδομένων *n* πόλεων με γνωστές μεταξύ τους αποστάσεις, να βρεθεί η συντομότερη διαδρομή που περνά από όλες τις πόλεις ακριβώς μία φορά πριν επιστρέψει στην αφετηρία.
- Εναλλακτικά: Βρες το *Hamiltonian κύκλωμα σε ένα ζυγισμένο συνδεδεμένο γράφο*
- Παράδειγμα:



Ο περιοδεύων πωλητής με εξαντλητική αναζήτηση

Διαδρομή	Κόστος
a→b→c→d→a	2+3+7+5 = 17
a→b→d→c→a	2+4+7+8 = 21
a→c→b→d→a	8+3+4+5 = 20
a→c→d→b→a	8+7+4+2 = 21
a→d→b→c→a	5+4+3+8 = 20
a→d→c→b→a	5+7+3+2 = 17

- Αποδοτικότητα

Παράδειγμα 2: πρόβλημα του σάκου

<u>Πρόβλημα</u>: Δεδομένων n αντικειμένων με βάρη w_1, w_2, …,w_n και αξίες v_1,v_2,…,v_n και ενός σάκου χωρητικότητας W, να βρεθεί το πολυτιμότερο υποσύνολο των αντικειμένων που ταιριάζουν στο σάκο

<u>Παράδειγμα</u>: *χωρητικότητα* W=16

Είδος	Βάρος	Αξία ($)
1	2	20
2	5	30
3	10	50
4	5	10

Σάκος με εξαντλητική αναζήτηση

Υποσύνολο	Συνολικό Βάρος	Συνολική Αξία ($)	Υποσύνολο	Συνολικό Βάρος	Συνολική Αξία ($)
{1}	2	20	{2, 4}	10	40
{2}	5	30	{3, 4}	15	60
{3}	10	50	{1, 2, 3}	17	not feasible
{4}	5	10	{1, 2, 4}	12	60
{1, 2}	7	50	{1, 3, 4}	17	not feasible
{1, 3}	12	70	{2, 3, 4}	20	not feasible
{1, 4}	7	30	{1, 2, 3, 4}	22	not feasible
{2, 3}	15	80			

Ανάθεση με εξαντλητική αναζήτηση

Πρόβλημα: Δεδομένων *n* ατόμων και *n* εργασιών, όπου *C[i,j]* είναι το κόστος αν το άτομο *i* αναλάβει την εργασία *j, να* ανατεθεί ένα άτομο σε κάθε εργασία με συνολικό ελάχιστο κόστος

Παράδειγμα:

	Εργασία 1	**Εργασία 2**	**Εργασία 3**	**Εργασία 4**
Άτομο 1	**9**	**2**	**7**	**8**
Άτομο 2	**6**	**4**	**3**	**7**
Άτομο 3	**5**	**8**	**1**	**8**
Άτομο 4	**7**	**6**	**9**	**4**

Ανάθεση με εξαντλητική αναζήτηση

Ανάθεση	*Συνολικό κόστος*
<1,2,3,4>	9+4+1+4=18
<1,2,4,3>	9+4+8+9=30
<1,3,2,4>	9+3+8+4=24
<1,3,4,2>	9+3+8+6=26
<1,4,2,3>	9+7+8+9=33
<1,4,3,2>	9+7+1+6=23
…………	…………….

Αποδοτικότητα;

Τελικά σχόλια

- Οι αλγόριθμοι εξαντλητικής αναζήτησης τρέχουν σε ρεαλιστικούς χρόνους <u>μόνο για μικρά στιγμιότυπα</u>
- Κάποιες φορές υπάρχουν καλύτερες εναλλακτικές λύσεις:
  - Κυκλώματα Euler
  - Συντομότερα μονοπάτια
  - Ελάχιστα ζευγνύοντα δένδρα
  - Πρόβλημα ανάθεσης
- Μερικές φορές όμως η εξαντλητική αναζήτηση (ή κάποια παραλλαγή) είναι η μοναδική γνωστή λύση

Σημείωμα Αναφοράς

Copyright Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Ιωάννης Μανωλόπουλος, Αναστάσιος Γούναρης. «Αλγόριθμοι. ». Έκδοση: 1.0. Θεσσαλονίκη 2015. Διαθέσιμο από τη δικτυακή διεύθυνση: http://eclass.auth.gr/courses/OCRS417/

Σημείωμα Αδειοδότησης

Τέλος ενότητας

Επεξεργασία: Ανδρέας Κοσματόπουλος
Θεσσαλονίκη, Αύγουστος 2015

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

ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

Σημειώματα

Σημείωμα Ιστορικού Εκδόσεων Έργου

Το παρόν έργο αποτελεί την έκδοση 1.00.

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.