



## Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

### **Ανάλυση και Σχεδίαση Αλγορίθμων** **Εργασία 2 – Ασκήσεις Β**

#### **Ομάδα Υλοποίησης:**

Χανός Στέφανος – 1573

Συγκούνας Ιωάννης – 1556

#### **Άσκηση 1:**

Αρχικά ξεκινάμε στην `main` και ορίζουμε τις μεταβλητές που θα αξιοποιήσουμε για τις χρονομετρήσεις μας. Θέτουμε ως πίνακες τα στοιχεία  $n$  και  $k$  ώστε να τους ορίσουμε δέκα τιμές με βάση τις οποίες θα τρέξουμε το πρόγραμμα μας. Ζητάμε από τον χρήστη να διαλέξει με ποια συνάρτηση θα ήθελε να κάνει την ταξινόμηση των στοιχείων του και κάνουμε κατάλληλο έλεγχο για την ορθή απάντηση του.

Με την πρώτη επιλογή καλείται η αναδρομική συνάρτηση μας `permutationCoeff1` που δίνεται από την εκφώνηση της άσκησης. Την καλούμε δέκα φορές ανάλογα με τις δέκα τιμές των  $n$  και  $k$  μας και χρονομετρούμε κάθε μια από αυτές, αποθηκεύουμε τις μετρήσεις μας σε έναν πίνακα και από τον συνολικό χρόνο βγάζουμε έναν μέσο όρο για να δούμε πόσο αποτελεσματική είναι η συνάρτηση. Έχει πολυπλοκότητα  $n + n * k$  άρα  $n$ .

```
Would you like to use the recursive function (1) , the Bottom-Up function (2) or the Top-Down one (3)
??
1
Recursive
If you have 4 elements and you want to arrange 1 of them , there are 4 possible ways
Time needed [0.000000]
If you have 25 elements and you want to arrange 11 of them , there are 1829115904 possible ways
Time needed [0.049000]
If you have 36 elements and you want to arrange 9 of them , there are 543574016 possible ways
Time needed [0.814000]
If you have 5 elements and you want to arrange 30 of them , there are 0 possible ways
Time needed [0.000000]
If you have 31 elements and you want to arrange 13 of them , there are 1394025472 possible ways
Time needed [2.199000]
If you have 3 elements and you want to arrange 0 of them , there are 1 possible ways
Time needed [0.000000]
If you have 41 elements and you want to arrange 9 of them , there are 1640905600 possible ways
Time needed [2.968000]
If you have 37 elements and you want to arrange 11 of them , there are 1864562688 possible ways
Time needed [7.741000]
If you have 56 elements and you want to arrange 8 of them , there are 932211840 possible ways
Time needed [11.093000]
If you have 49 elements and you want to arrange 10 of them , there are 309180416 possible ways
Time needed [66.070000]
Average time for the [RECURSIVE] function is: 9.093400
```

Με την δεύτερη επιλογή καλείται η Bottom-Up υλοποίηση permutationCoeff2 η οποία έχει ακριβώς το ίδιο σκεπτικό για τις χρονομετρήσεις και τις εκτυπώσεις με την αναδρομική. Ορίζουμε έναν πίνακα μέσα στην συνάρτηση στον οποίο αποθηκεύουμε τα αποτελέσματα μας. Άμα το j είναι ίσο με το 0 βάζουμε στην αντίστοιχη θέση του πίνακα την τιμή 1 , αν το j είναι μεγαλύτερο του i τότε βάζουμε στην αντίστοιχη θέση του πίνακα την τιμή 0 καθώς δεν μπορεί να ταξινομηθεί και σε οποιαδήποτε άλλη περίπτωση υπολογίζουμε τους πιθανούς τρόπους ταξινόμησης και τοποθετούμε το αποτέλεσμα μας στην αντίστοιχη θέση του πίνακα.

```

Would you like to use the recursive function (1) , the Bottom-Up function (2) or the Top-Down one (3)
??
2
Bottom-Up
If you have 4 elements and you want to arrange 1 of them , there are 4 possible ways
Time needed [0.000000]
If you have 25 elements and you want to arrange 11 of them , there are 1829115904 possible ways
Time needed [0.000000]
If you have 36 elements and you want to arrange 9 of them , there are 543574016 possible ways
Time needed [0.000000]
If you have 5 elements and you want to arrange 30 of them , there are 0 possible ways
Time needed [0.000000]
If you have 31 elements and you want to arrange 13 of them , there are 1394025472 possible ways
Time needed [0.000000]
If you have 3 elements and you want to arrange 0 of them , there are 1 possible ways
Time needed [0.000000]
If you have 41 elements and you want to arrange 9 of them , there are 1640905600 possible ways
Time needed [0.000000]
If you have 37 elements and you want to arrange 11 of them , there are 1864562688 possible ways
Time needed [0.000000]
If you have 56 elements and you want to arrange 8 of them , there are 932211840 possible ways
Time needed [0.000000]
If you have 49 elements and you want to arrange 10 of them , there are 309180416 possible ways
Time needed [0.000000]
Average time for the [Bottom-Up] function is: 0.000000

```

Τέλος με την τρίτη επιλογή καλείται η Top-Down υλοποίηση permutationCoeff3 η οποία έχει ακριβώς το ίδιο σκεπτικό για τις χρονομετρήσεις και τις εκτυπώσεις με την αναδρομική. Για την υλοποίησή της αξιοποιούμε έναν πίνακα τον οποίο έχουμε αρχικοποιήσει με μέγεθος , αυτό των μεγαλύτερων τιμών των  $n$  και  $k$  και τον έχουμε γεμίσει με -1 καλώντας την συνάρτηση Initialization. Στην περίπτωση που η θέση του πίνακα έχει τιμή -1 (άρα δεν έχει γίνει κάποιος υπολογισμός ακόμα) μπαίνουμε μέσα στην if μας όπου άμα το  $k$  είναι ίσο με το 0 βάζουμε στην αντίστοιχη θέση του πίνακα την τιμή 1 , αν το  $k$  είναι μεγαλύτερο του  $n$  τότε βάζουμε στην αντίστοιχη θέση του πίνακα την τιμή 0 καθώς δεν μπορεί να ταξινομηθεί και σε οποιαδήποτε άλλη περίπτωση υπολογίζουμε τους πιθανούς τρόπους ταξινόμησης και τοποθετούμε το αποτέλεσμα μας στην αντίστοιχη θέση του πίνακα.

```

Would you like to use the recursive function (1) , the Bottom-Up function (2) or the Top-Down one (3)
??
3
Top-Down
If you have 4 elements and you want to arrange 1 of them , there are 4 possible ways
Time needed [0.000000]
If you have 25 elements and you want to arrange 11 of them , there are 1829115904 possible ways
Time needed [0.000000]
If you have 36 elements and you want to arrange 9 of them , there are 543574016 possible ways
Time needed [0.000000]
If you have 5 elements and you want to arrange 30 of them , there are 0 possible ways
Time needed [0.000000]
If you have 31 elements and you want to arrange 13 of them , there are 1394025472 possible ways
Time needed [0.000000]
If you have 3 elements and you want to arrange 0 of them , there are 1 possible ways
Time needed [0.000000]
If you have 41 elements and you want to arrange 9 of them , there are 1640905600 possible ways
Time needed [0.000000]
If you have 37 elements and you want to arrange 11 of them , there are 1864562688 possible ways
Time needed [0.000000]
If you have 56 elements and you want to arrange 8 of them , there are 932211840 possible ways
Time needed [0.000000]
If you have 49 elements and you want to arrange 10 of them , there are 309180416 possible ways
Time needed [0.000000]
Average time for the [Top-Down] function is: 0.000000

```

## Άσκηση 2:

Αρχικά ξεκινάμε στην main όπου ζητάμε από τον χρήστη να εισάγει τις διαστάσεις του πίνακα. Δηλώνουμε τον πίνακα με αυτές τις διαστάσεις , τον γεμίζουμε με τυχαίους αριθμούς από το 0 μέχρι το 9 και τον εκτυπώνουμε. Στην συνέχεια καλούμε την συνάρτησή μας fun() για να πάρουμε το ελάχιστο κόστος από την αρχή ως το τέλος και το μονοπάτι αυτού.

Συγκεκριμένα , ορίζουμε τρεις πίνακες. Γεμίζουμε τους δυο καταλληλά για να μην βγαίνουμε εκτός ορίων στην συνεχεία. Ο MinCost[][] γεμίζει με το άθροισμα του κόστους κάθε κίνησης από μια θέση στην επόμενη της. Παράλληλα ο πίνακας Position[][] γεμίζει με τις κατάλληλες τιμές οι οποίες δηλώνουν από προήλθε η κίνηση μας , με βάση το ποια κίνηση είχε το μικρότερο κόστος. Τέλος για να καταφέρουμε να τυπώσουμε και το μονοπάτι αξιοποιούμε τον πίνακα Path[][] τον οποίο αρχικοποιούμε στην τελική θέση του πίνακα και με βάση το τι τιμή έχει ο πίνακας Position[][] κάνουμε την κατάλληλη αφαίρεση και βρίσκουμε το μονοπάτι από το τέλος προς την αρχή.

```
Please enter the Width of the table
3
Please enter the Height of the table
3
-----
[2][3][0]
[6][7][7]
[3][6][7]
-----
The minimum-cost path is:
(0,0)
(0,1)
(0,2)
(1,2)
(2,2)
The cost is: [19]
```

```
Please enter the Width of the table
5
Please enter the Height of the table
5
-----
[6][8][9][8][5]
[1][5][4][8][4]
[1][3][9][1][4]
[8][6][0][9][7]
[5][9][3][9][4]
-----
The minimum-cost path is:
(0,0)
(1,0)
(2,0)
(2,1)
(3,1)
(3,2)
(4,2)
(4,3)
(4,4)
The cost is: [33]
```

Please enter the Width of the table  
10  
Please enter the Height of the table  
10

-----  
[7][1][0][8][2][0][1][5][1][7]  
[2][5][9][6][4][4][2][5][2][2]  
[3][5][2][9][9][0][1][5][1][2]  
[0][2][1][2][2][9][4][1][1][8]  
[2][1][1][2][3][3][0][6][0][8]  
[7][5][7][4][2][1][2][1][5][8]  
[9][1][6][1][2][7][2][9][8][2]  
[2][4][2][1][7][6][8][3][3][0]  
[4][1][5][8][4][7][5][1][2][1]  
[6][3][3][4][6][3][9][1][6][8]  
-----

The minimum-cost path is:

(0,0)  
(1,0)  
(2,0)  
(3,0)  
(4,0)  
(4,1)  
(4,2)  
(4,3)  
(4,4)  
(5,4)  
(5,5)  
(5,6)  
(6,6)  
(7,6)  
(7,7)  
(8,7)  
(8,8)  
(8,9)  
(9,9)

The cost is: [51]

Please enter the Width of the table

20

Please enter the Height of the table

20

-----  
[6][7][8][5][5][6][0][6][6][8][3][2][5][2][0][0][1][4][3][4]  
[7][2][5][2][5][0][5][0][5][7][4][6][8][5][9][3][0][6][3][1]  
[0][2][2][2][9][3][0][4][3][5][5][1][8][1][3][2][3][0][3][5]  
[2][7][9][8][3][8][6][5][9][1][4][9][8][0][4][9][3][1][2][1]  
[7][8][0][3][7][0][1][3][6][5][5][2][9][3][2][1][8][6][5][2]  
[4][2][1][8][6][1][8][3][8][0][4][5][9][3][5][1][0][5][5][2]  
[2][8][9][0][3][6][8][2][1][5][6][9][6][0][8][8][9][9][9][2]  
[3][1][1][7][4][5][4][0][2][8][5][9][6][3][6][7][5][2][4][8]  
[5][2][2][0][3][9][8][2][9][0][7][7][5][9][7][0][8][8][8][7]  
[9][0][0][0][8][0][3][9][4][9][1][1][7][2][2][4][8][7][2][5]  
[2][0][0][1][1][3][7][2][3][1][9][4][4][7][4][4][9][2][5][3]  
[3][4][4][5][3][1][7][6][2][0][4][8][2][4][7][9][5][1][0][8]  
[3][4][3][5][4][5][6][8][2][9][9][8][4][7][2][8][9][7][2][4]  
[9][8][8][9][9][8][0][9][5][5][9][7][0][7][5][8][5][1][9][8]  
[8][9][2][7][5][5][1][0][3][4][9][9][5][3][3][6][9][2][2][3]  
[7][8][5][8][6][5][8][9][9][7][8][7][2][9][7][4][3][2][4][3]  
[3][1][9][5][0][9][9][4][8][9][9][5][4][5][8][7][7][2][5][4]  
[1][6][2][5][4][3][6][7][2][0][6][5][2][4][1][6][8][1][7][6]  
[2][0][9][1][6][3][5][8][2][6][1][6][0][9][0][5][1][7][2][0]  
[8][6][0][8][2][3][6][3][4][0][3][8][4][1][6][5][1][2][2][6]

-----  
The minimum-cost path is:

(0,0)  
(1,0)  
(2,0)  
(3,0)  
(4,0)  
(5,0)  
(6,0)  
(7,0)  
(7,1)  
(8,1)  
(9,1)  
(10,1)  
(10,2)  
(10,3)  
(10,4)  
(10,5)  
(10,6)  
(10,7)  
(10,8)  
(10,9)  
(11,9)  
(11,10)  
(11,11)  
(11,12)  
(12,12)  
(13,12)  
(14,12)  
(15,12)  
(16,12)  
(17,12)  
(17,13)  
(17,14)  
(18,14)  
(18,15)  
(18,16)  
(19,16)  
(19,17)  
(19,18)  
(19,19)

The cost is: [105]