Θεμελιώδη Θέματα Επιστήμης Υπολογιστών 2021-22

2η σειρά γραπτών ασκήσεων

(αλγοριθμικές τεχνικές – αριθμητικοί αλγόριθμοι αλγόριθμοι γράφων – δυναμικός προγραμματισμός)

Άσκηση 1.

Α) Για τους πύργους του Hanoi γνωρίζουμε τα εξής:

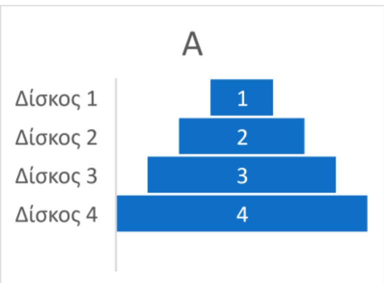
- Σε μια κίνηση μπορούμε να μετακινήσουμε έναν μόνο δίσκο.
- Μπορούμε να μετακινήσουμε μόνο τον ανώτερο δίσκο μιας στοίβας κι όχι κάποιον ενδιάμεσο.
- Ένας δίσκος με μεγαλύτερη διάμετρο δεν μπορεί να τοποθετηθεί πάνω από έναν δίσκο με μικρότερη διάμετρο.

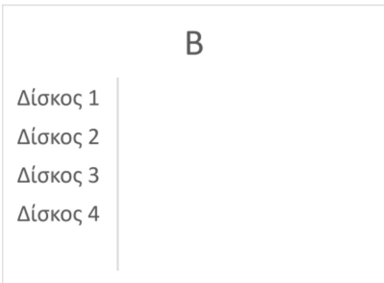
Για την υλοποίηση του προγράμματος θα σκεφτούμε τα εξής:

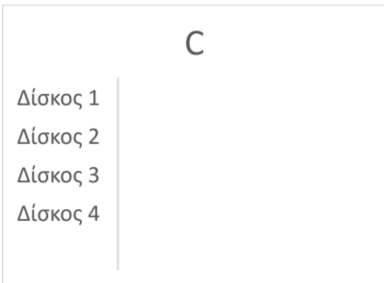
- Διαθέτουμε 3 ράβδους και n δίσκους. Αριθμούμε τους δίσκους από κάτω προς τα πάνω, δηλαδή ο δίσκος με τη μεγαλύτερη διάμετρο είναι ο n-οστός δίσκος, ενώ ο δίσκος με τη μικρότερη διάμετρο είναι ο δίσκος #1. Επίσης ονομάζουμε τις 3 ράβδους ως A, B, C.
- Καθώς ο σκοπός του παιχνιδιού των πύργων του Hanoi είναι να μεταφέρουμε όλους τους δίσκους μιας στοίβας από μια ράβδο σε μια άλλη, μπορούμε να πούμε πως κάθε ράβδος έχει και έναν σκοπό: origin, destination, empty. Έτσι, στο πρόγραμμά μας, κάθε ράβδο rod θα την ονομάσουμε, αντίστοιχα: origin_rod, destination_rod, empty_rod.
- Αν το n = 1, τότε δεν έχουμε παρά να μετακινήσουμε τον δίσκο από το origin_rod στο destination_rod και να τυπώσουμε το αντίστοιχο μήνυμα. Συνεπώς έχουμε μια κίνηση, άρα H(1)=1.
- Αν το n > 1, η κατάσταση είναι λίγο πιο περίπλοκη. Σε κάθε βήμα θα καλούμε αναδρομικά τη συνάρτησή μας για n-1 δίσκους έτσι, ώστε το origin_rod να παραμένει το ίδιο, αλλά το empty_rod και το destination_rod να εναλλάσσονται μεταξύ τους. Έπειτα, τυπώνεται το αντίστοιχο μήνυμα και καλείται η συνάρτησή μας πάλι για n-1, αλλά με origin_rod το αρχικό empty_rod, με empty_rod το αρχικό origin_rod και με το αρχικό destination_rod. Σε αυτή την περίπτωση έχουμε H(n-1) κινήσεις για τη μετακίνηση των n-1 δίσκων, μια επιπλέον για την τοποθέτηση του n-οστού δίσκου στο destination_rod και στη συνέχεια άλλες H(n-1) κινήσεις για την επανατοποθέτηση των n-1 δίσκων πάνω στον n-οστό δίσκο. Άρα H(n) = 2*H(n-1)+1

Ας δώσουμε ένα παράδειγμα, για n = 4, origin_rod = A, empty_rod = B, destination_rod = C:

Αρχικά οι πύργοι του Hanoi έχουν την εξής μορφή:



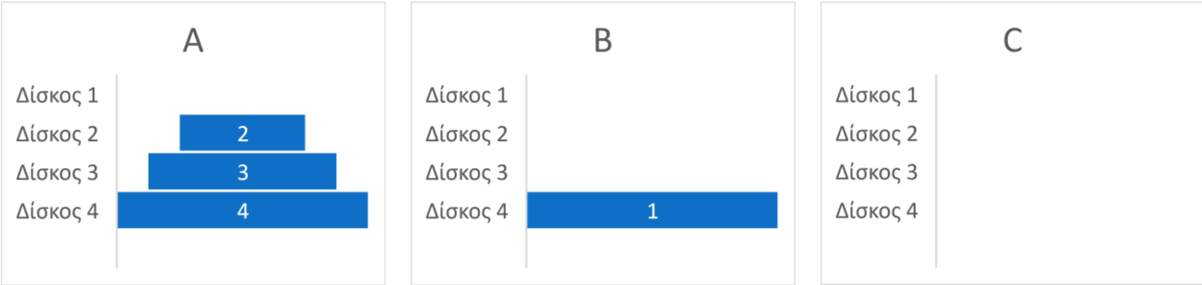


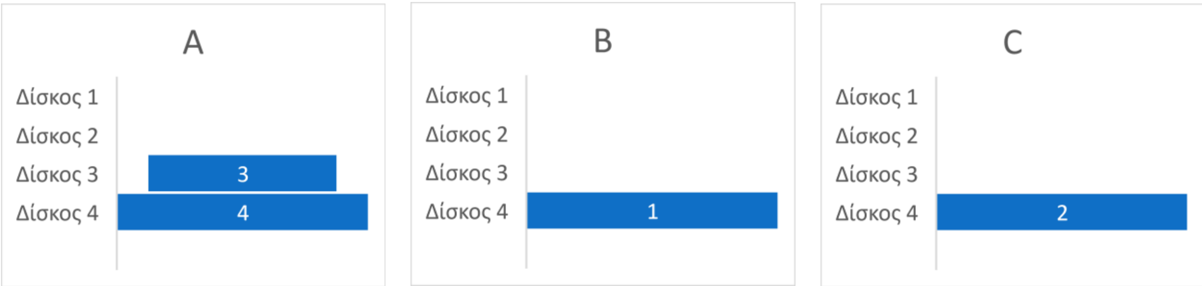


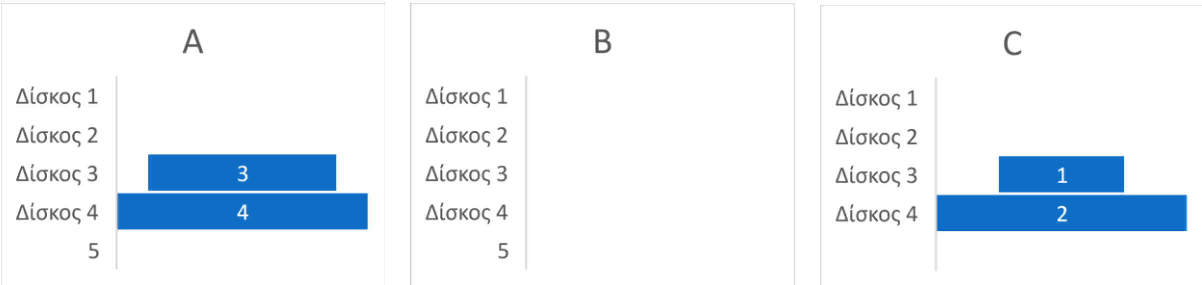
Εφαρμόζοντας τα παραπάνω βήματα, η συνάρτησή μας θα κληθεί για

1. n-1 = 3, origin_rod = A, empty_rod = C, destination_rod = B, και στη συνέχεια n-1 = 3, origin_rod = B, empty_rod = A, destination_rod = C.
2. n-1 = 2, origin_rod = A, empty_rod = B, destionation_rod = C και στη συνέχεια n-1 = 2, origin_rod = C, empty_rod = A, destination_rod = B.
3. n-1 = 1, origin_rod = A, empty_rod = C, destination_rod = B και στη συνέχεια n-1 = 1, origin_rod = B, empty_rod = A, destination_rod = C.

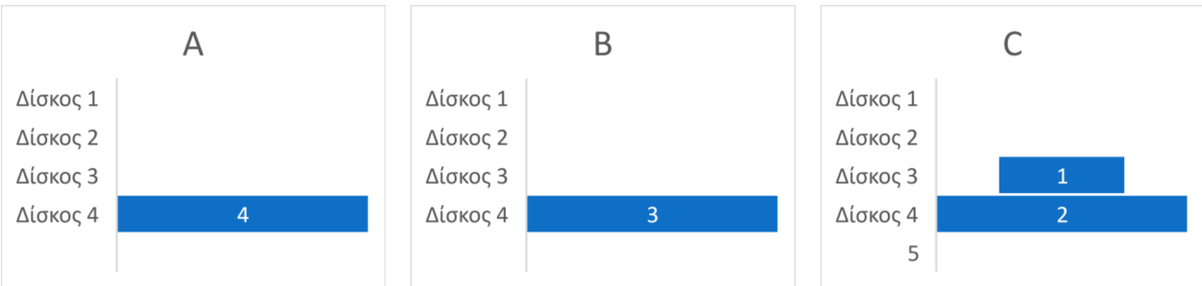
Θα οπτικοποιήσουμε τον παρπάνω αλγόριθμο, ξεκινώντας από το βήμα 3, εφόσον είναι αναδρομικός.

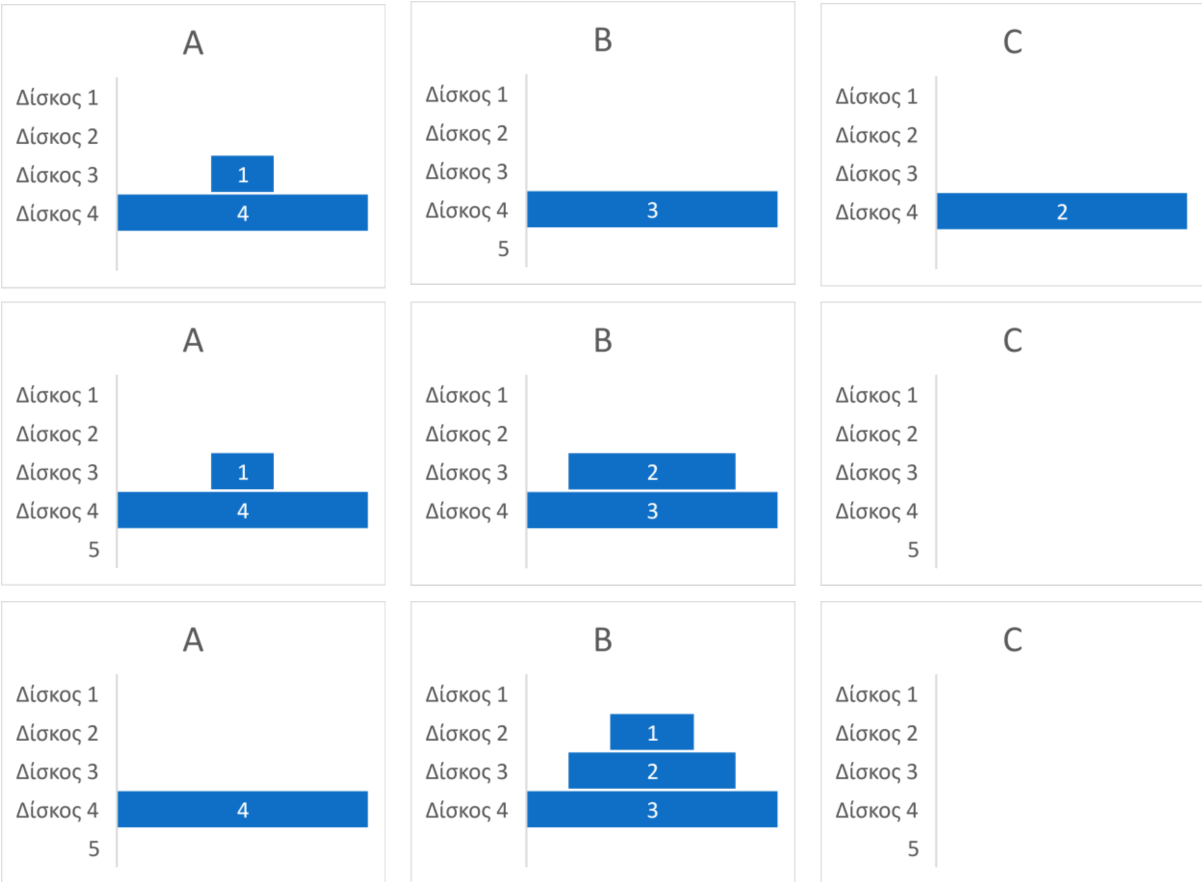




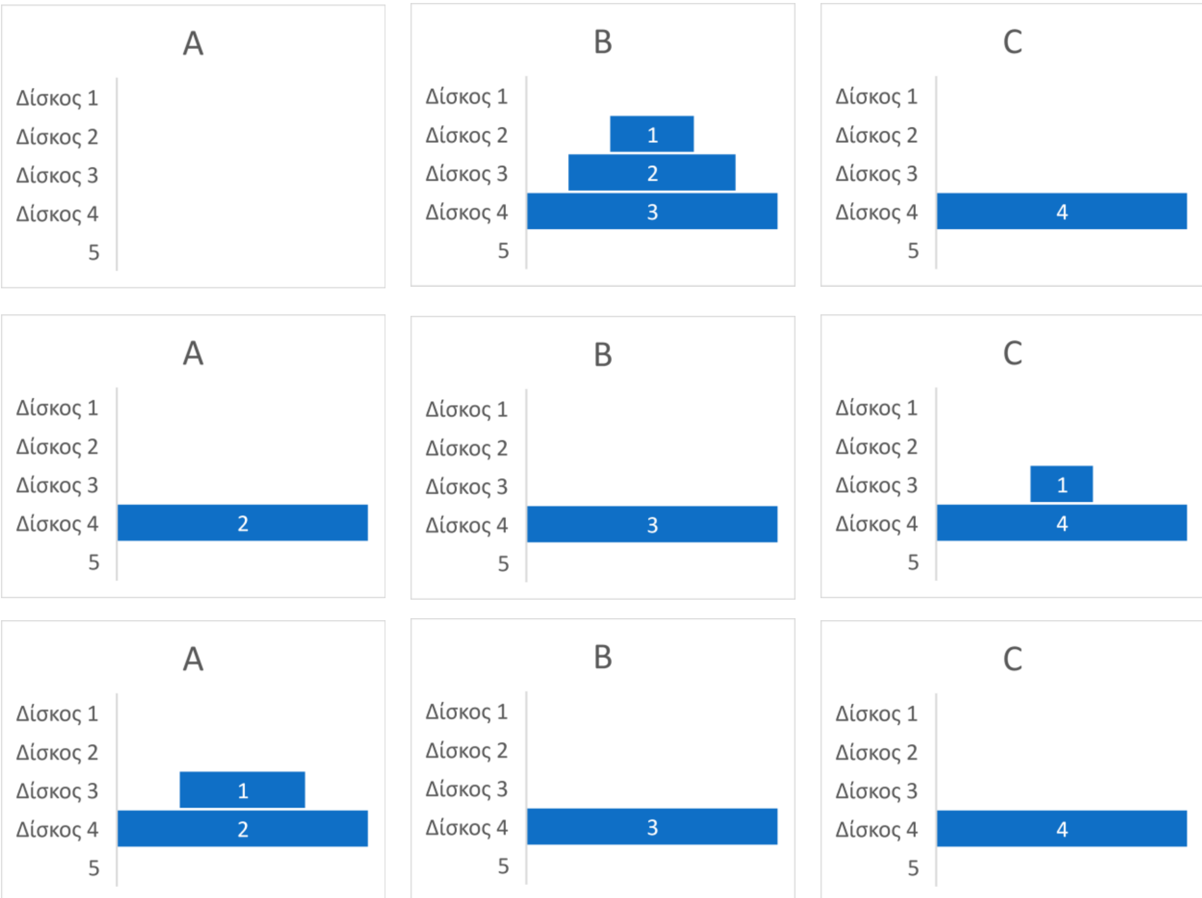


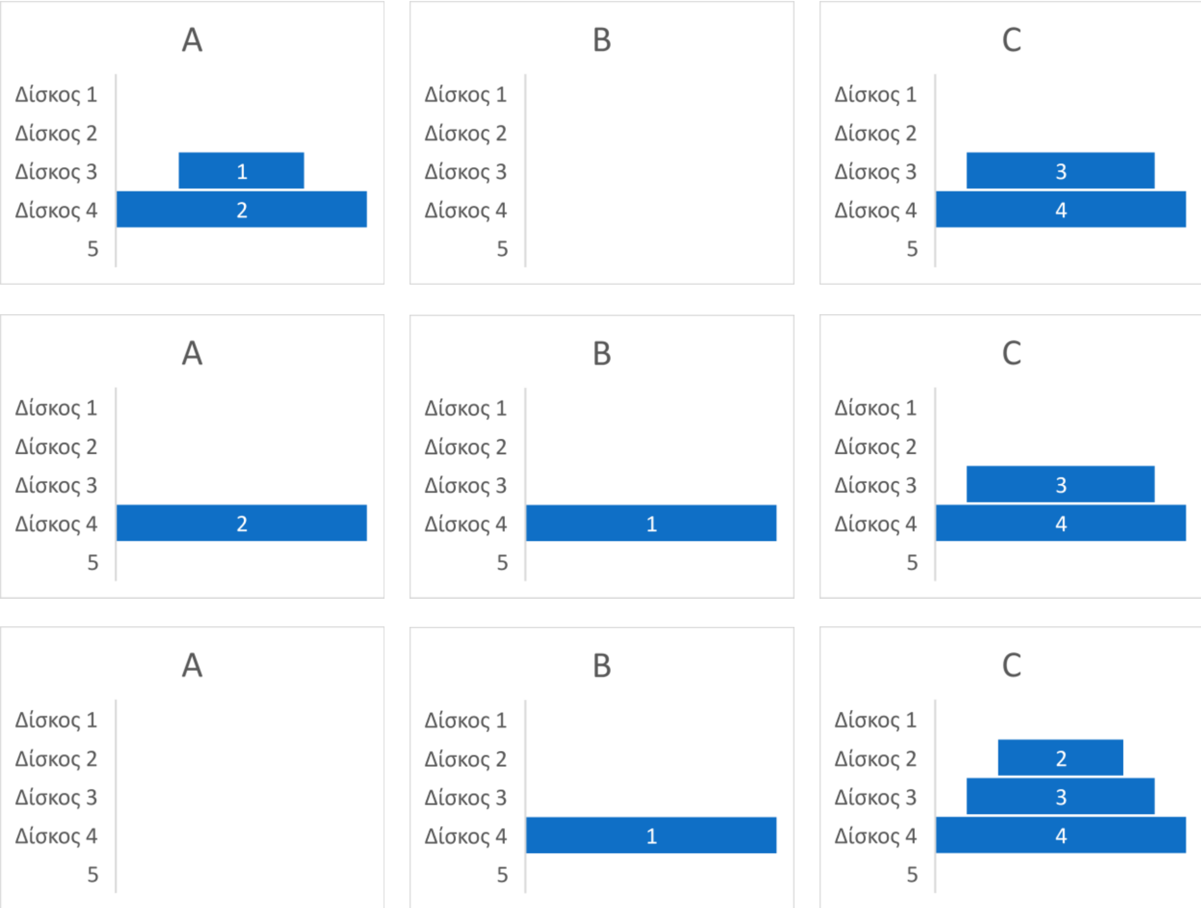
Συνεχίζουμε με το βήμα 2.



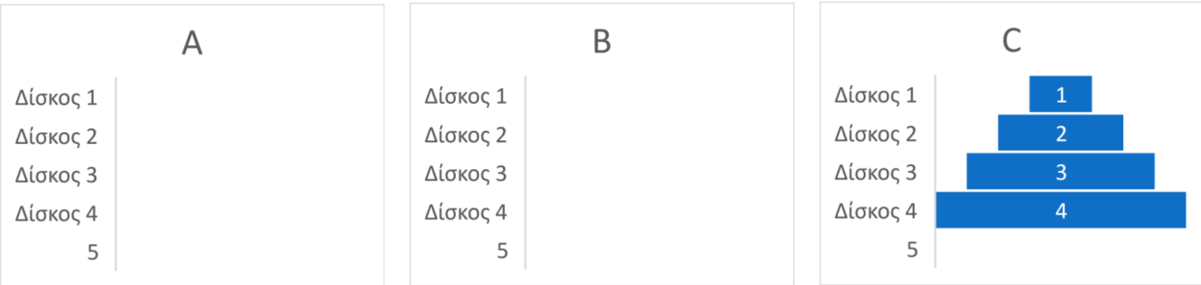


Συνεχίζουμε με το βήμα 1:





Τελικά καταλήγουμε στο ζητούμενο, το οποίο είναι:



Συνεπώς ο αλγόριθμός μας δουλεύει. Στη c++ γράφεται ως εξής:

```
Run  Debug  Stop  Share  Save  { } Beautify
ain.cpp

#include <iostream>
using namespace std;

void TowerOfHanoi(int n, char origin_rod, char empty_rod, char destination_rod) {
    if (n == 1) {
        printf("\nΜετακινούμε τον δίσκο 1 από τη ράβδο %c στη ράβδο %c.", origin_rod, destination_rod);
        return;
    }
    TowerOfHanoi(n - 1, origin_rod, destination_rod, empty_rod);
    printf("\nΜετακινούμε τον δίσκο %d από τη ράβδο %c στη ράβδο %c.", n, origin_rod, destination_rod);
    TowerOfHanoi(n - 1, empty_rod, origin_rod, destination_rod);
}

int main() {
    int A, B, C, n;
    cin >> n;
    TowerOfHanoi(n, A, B,C);
}
```