

Ενότητα 1: Εισαγωγή στον Α/Σ Προγραμματισμό

## Διάλεξη 3 - 30/1/24

Κλάσεις και Αντικείμενα  
Διατύπωση και Επίλυση Υπολογιστικών Προβλημάτων



## Ανασκόπηση



- Στο προηγούμενο μάθημα παρουσιάσαμε τον κύκλο ζωής (lifecycle) προγραμμάτων Java.

## Περίγραμμα



- Επίλυση Υπολογιστικών Προβλημάτων
- Παραδείγματα και Κατηγορίες Υπολογιστικών Προβλημάτων:
  - ▶ Προβλήματα Βελτιστοποίησης
  - ▶ **Προβλήματα Αναπαράστασης Χώρου Καταστάσεων**
- «Καλώς ορισμένα» προβλήματα



# Κλάσεις και Αντικείμενα

ΠΩΣ ΜΠΟΡΟΥΜΕ ΝΑ  
ΕΠΑΝΑΧΡΗΣΙΜΟΠΟΙΗΣΟΥΜΕ  
ΤΑ ΠΡΟΓΡΑΜΜΑΤΑ ΑΥΤΑ;

```
/*
 * Created by mdd on 15/01/2017.
 */
public class LeapYear {

    public static void main(String[] args) {
        int year = Integer.parseInt(args[0]);

        // divisible by 4
        boolean isLeapYear = (year % 4 == 0);

        // divisible by 4 and not 100
        isLeapYear = isLeapYear && (year % 100 != 0);

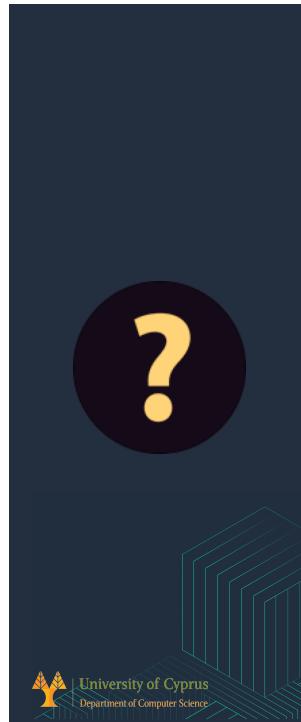
        // divisible by 4 and not 100 unless divisible by 400
        isLeapYear = isLeapYear || (year % 400 == 0);

        System.out.println(isLeapYear);
    }
}
```

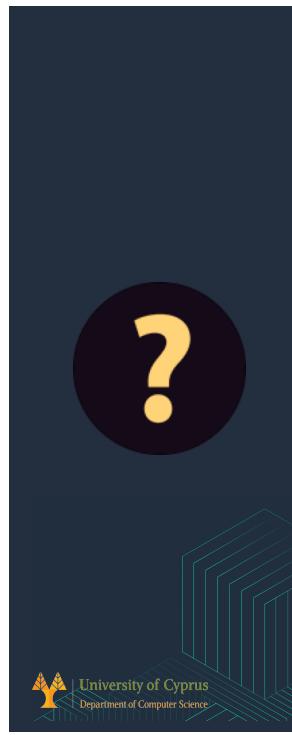
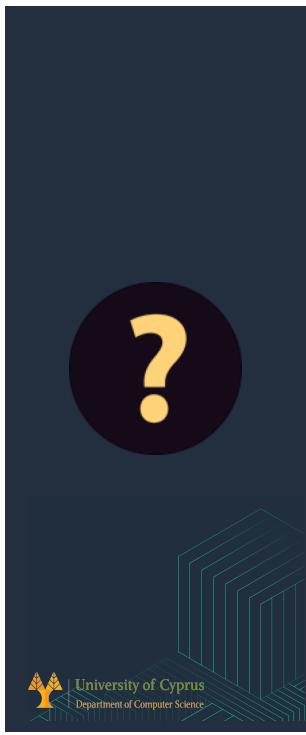
- Πώς συντάσσονται;
- Πως εκτελούνται;
- Πώς χρησιμοποιούνται;
- Τι **μειονεκτήματα** έχουν;

*M. Δικαιάκος, ΕΠΛ133*





ΧΡΗΣΙΜΟΠΟΙΗΣΑΜΕ  
ΑΝΤΙΚΕΙΜΕΝΑ;  
  
JAVA ΧΩΡΙΣ  
ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΕΙΑ!



## Κλάσεις στις Γλώσσες Προγραμματισμού

- Η κλάση στη Java είναι μια «μήτρα»/«μηχάνημα» παραγωγής αντικειμένων
- Η Java μας επιτρέπει να ορίζουμε στα προγράμματα μας τέτοιες «μήτρες» και από αυτές να «παράγουμε» αντικείμενα στα οποία αποθηκεύουμε τα δεδομένα των προγραμμάτων.

M. Δικαιάκος, ΕΠΛ133

## Κλάσεις

- Το βασικό συστατικό στοιχείο των προγραμμάτων JAVA είναι οι **κλάσεις** (class)
- Τι σημαίνει κλάση;
  - «Το σύνολο προσώπων ή πραγμάτων που συγκροτούν ένα σύνολο»
  - «Συλλογή μεταβλητών και συναρτήσεων»
  - (Συν.) Τάξη, κατηγορία, ομάδα
- **Ιδεότυπος:** αφηρημένο πρότυπο κάθε συγκεκριμένου συλλογικού προτύπου ή μορφής



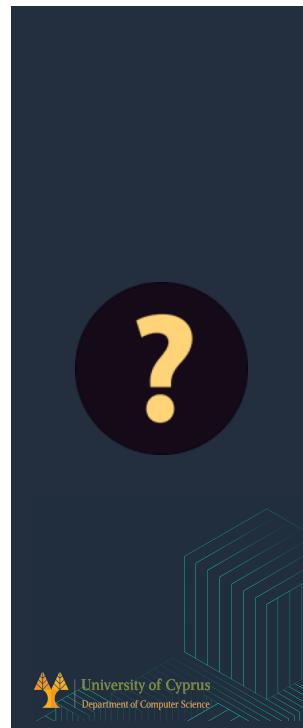
90

## Κλάσεις και Τύποι Δεδομένων

- Συνήθεις τύποι δεδομένων («αρχέγονοι»): integer, float, double, char, boolean

## Οι Κλάσεις σαν Τύποι Δεδομένων

- Οι κλάσεις είναι προγραμματιστικώς οριζόμενοι «τύποι δεδομένων».
- Καθορίζουν:
  - περιεχόμενο
  - σύνταξη
  - σημασία
- των **τιμών** (values) που παράγονται από/υπάγονται σε αυτές
- Συμπεριλαμβάνουν και **λειτουργικότητα** («συμπεριφορά») - υλοποιημένη με **κώδικα**, όπως και  **έλεγχο πρόσβασης** σε πεδία δεδομένων κλπ.
- Βασική **προγραμματιστική δομή** του Α/Σ Προγραμματισμού.



ΤΙ ΧΑΡΑΚΤΗΡΙΖΕΙ ΕΝΑ ΤΥΠΟ  
ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΤΙ ΤΟΝ  
ΔΙΑΦΟΡΟΠΟΙΕΙ ΑΠΟ  
ΚΑΠΟΙΟΝ ΆΛΛΟ;

## Κλάσεις και Αντικείμενα

- Οι κλάσεις χρησιμοποιούνται κυρίως για να «παράξουμε» αντικείμενα (τιμές)
- Τα **Αντικείμενα** είναι η βασική **Δομή Δεδομένων** στον Α/Σ Προγραμματισμό





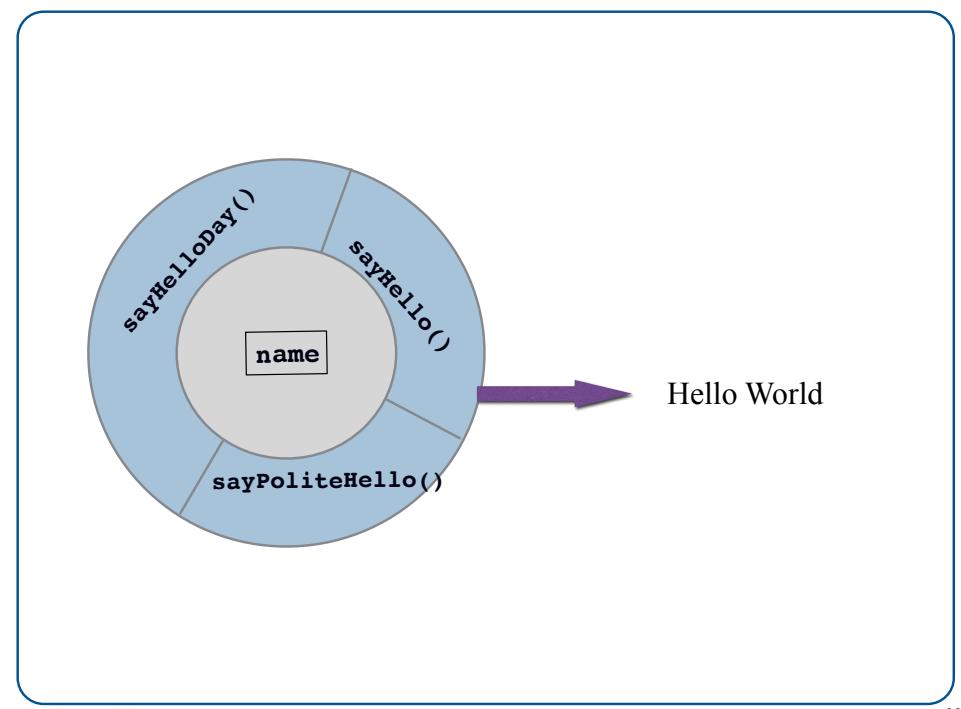
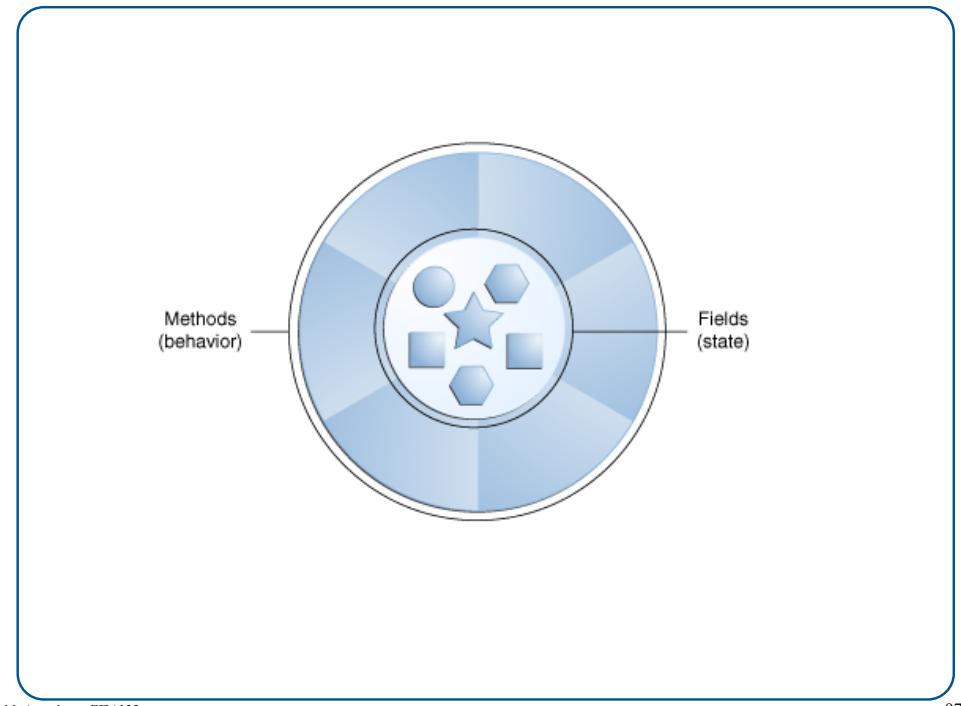
Οι υπολογισμοί που πραγματοποιούνται κατά την εκτέλεση ενός αντικειμενοστρεφούς προγράμματος για την επίλυση ενός προβλήματος συνιστούν μια αλληλουχία αλληλεπιδράσεων μεταξύ αντικειμένων που εξυπηρετούν το ένα το άλλο.

## ΠΡΟΒΛΗΜΑ

Εκτύπωσε το μήνυμα  
“Γειά σου κόσμε”



1. Γράψε μια κλάση που να εκτυπώνει έναν χαιρετισμό
2. Κάνε χρήση της κλάσης αυτής για να εκτυπώσεις το μήνυμα “Γειά σου κόσμε”



## Μια καλύτερη προσπάθεια...

```
public class Greeter {  
    public Greeter(String aName) {  
        name = aName;  
    }  
  
    public String sayHello() {  
        return "Hello, " + name + "!";  
    }  
  
    private String name;  
}
```

constructor: κατασκευαστής αντικειμένων

μέθοδος: υλοποίηση λειτουργικότητας αντικειμένων Greeter

πεδίο δεδομένων (field)

## Χρήση της Greeter

```
public class GreeterTester {  
    public static void main(String[] args) {  
  
        Greeter worldGreeter = new Greeter("World");  
  
        String greeting = worldGreeter.sayHello();  
  
        System.out.println(greeting);  
    }  
}
```

## Τεκμηρίωση κώδικα

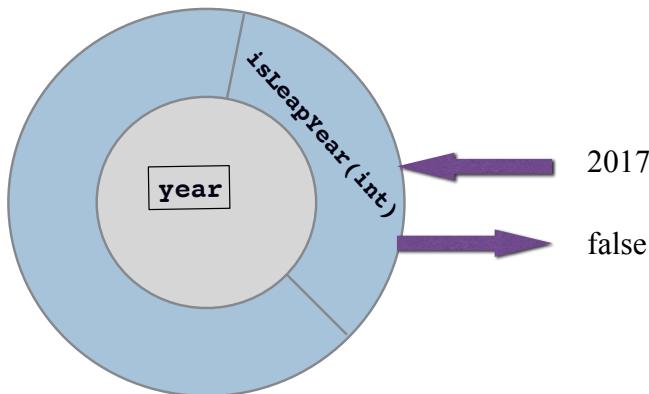
```
01: /**  
02:  A class for producing simple greetings.  
03: */  
04:  
05: public class Greeter {  
06:     /**  
07:      Constructs a Greeter object that can greet a person or entity.  
08:      @param aName the name of the person or entity who should  
09:      be addressed in the greetings.  
10:     */  
11:     public Greeter(String aName) {  
12:         name = aName;  
13:     }  
14:  
15:     /**  
16:      Greet with a "Hello" message.  
17:      @return a message containing "Hello" and the name of  
18:      the greeted person or entity.  
19:     */  
20:     public String sayHello() {  
21:         return "Hello, " + name + "!";  
22:     }  
23:  
24:     private String name;  
25: }
```

## ΠΡΟΒΛΗΜΑ:

Βρες αν ένα έτος είναι δίσεκτο



1. Γράψε μια κλάση που να περιέχει μια μέθοδο, η οποία δέχεται σαν παράμετρο ένα έτος ελέγχει αν είναι δίσεκτο
2. Κάνε χρήση της κλάσης αυτής για να διαβάζεις από το πληκτρολόγιο ένα έτος και να λαμβάνεις στην έξοδο πληροφόρηση για το αν είναι δίσεκτο ή όχι



```
public class TestLeapYear {
    public static void main(String[] args) {
        int year = Integer.parseInt(args[0]);
        LeapYear leapYearCheck = new LeapYear();
        System.out.println(leapYearCheck.isLeapYear(year));
    }
}
```

```
public class LeapYear {
    public boolean isLeapYear(int year) {
        boolean isLeapYear;

        // divisible by 4
        isLeapYear = (year % 4 == 0);

        // divisible by 4 and not 100
        isLeapYear = isLeapYear && (year % 100 != 0);

        // divisible by 4 and not 100 unless divisible by 400
        isLeapYear = isLeapYear || (year % 400 == 0);

        return isLeapYear;
    }
}
```

## Πόσα πρέπει να ξέρω;

- Υπάρχουν βασικοί αλγόριθμοι, η χρήση των οποίων εμφανίζεται σχεδόν σε κάθε πρόγραμμα (κυρίως αλγόριθμοι αναζήτησης και ταξινόμησης).
- Οι δομές δεδομένων που χρησιμοποιούνται συνήθως είναι γνωστές, ή προκύπτουν από τη σύνθεση γνωστών δομών:
  - Πίνακες (arrays)
  - Λίστες (lists)
  - Πίνακες κατακερματισμού (hash tables)
  - Δένδρα (trees)
- Προγραμματιστικές τεχνικές μας βοηθούν να κάνουμε τον κώδικα μας πιο γρήγορο, ευανάγνωστο, με λιγότερα σφάλματα.

## Περίγραμμα



- Κλάσεις και Αντικείμενα
- Επίλυση Υπολογιστικών Προβλημάτων
- Παραδείγματα και Κατηγορίες Υπολογιστικών Προβλημάτων
- «Καλώς ορισμένα» προβλήματα

M. D. Dikaiakos

## Επίλυση Προβλημάτων



### ΠΡΟΒΛΗΜΑ:

**Πες μου κάτι ενδιαφέρον για  
ένα έτος που σου δίνεται**

ΕΙΣΟΔΟΣ (input)

2020

Πρόγραμμα  
Java

ΕΞΟΔΟΣ (output)

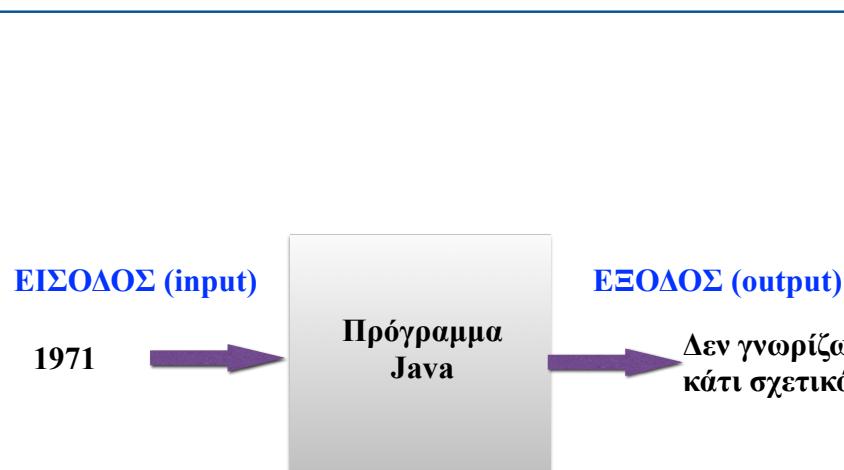
Δίσεκτο  
Covid-19



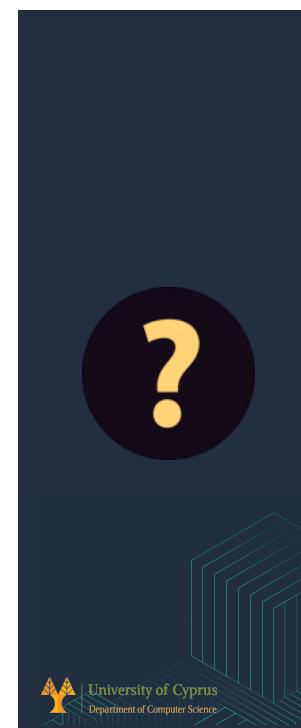
112



113



114



M. Δικαιάκος, ΕΠΛ133

 **ΒΡΕΙΤΕ ΛΥΣΗ!**

**ΜΠΟΡΕΙΤΕ ΝΑ  
ΕΠΑΝΑΧΡΗΣΙΜΟΠΟΙΗΣΕΤΕ  
ΤΟΝ ΚΩΔΙΚΑ ΤΗΣ LEAPYEAR  
ΠΟΥ ΔΕΙΞΑΜΕ ΠΡΙΝ;**

**ΠΩΣ ΘΑ ΛΥΣΕΤΕ ΤΟ  
ΠΡΟΒΛΗΜΑ;**

## Διαδικασία Επίλυσης Προβλημάτων

- **Ορισμός** και **ανάλυση** του προβλήματος
  - Αναγκαίες υποθέσεις, αφαιρετικότητα, υποπροβλήματα, τυπικός ορισμός
- Επινόηση πλήρους λύσης
- Διατύπωση λύσης σε βήματα με λεπτομέρεια (**αλγόριθμος**)
- **Έλεγχος** ορθότητας (testing), **αξιολόγηση** (evaluation) λύσης και **βελτίωση**
- **Τεκμηρίωση** (documentation) λύσης και **συντήρηση** (maintenance)

M. Δικαιάκος, ΕΠΛ133

116

## Επίλυση Υπολογιστικών Προβλημάτων

- **Θεμελίωση** (φορμαλιστική διατύπωση) Υπολογιστικού Προβλήματος: ορισμός και ανάλυση
  - Ποια είναι η **είσοδος** και ποια η **έξοδος**;
  - Ποιες άλλες πληροφορίες είναι απαραίτητες για τη λύση;
  - Ποια είναι η κατάλληλη αναπαράσταση του προβλήματος;
- Επινόηση και διατύπωση **αλγορίθμου** επίλυσης
  - Ποια είναι τα βήματα για επίλυση του προβλήματος;
- Κωδικοποίηση αλγορίθμου σε **πρόγραμμα**
- **Μεταγλώττιση**, **έλεγχος ορθότητας** και **αποσφαλμάτωση** προγράμματος
- **Τεκμηρίωση** και **συντήρηση** του προγράμματος

M. Δικαιάκος, ΕΠΛ133

117

## Αλγόριθμοι και Δομές Δεδομένων

**ΠΡΟΓΡΑΜΜΑΤΑ = ΑΛΓΟΡΙΘΜΟΙ + ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ** (Niklaus Wirth)

- Η ικανότητα να επιλύουμε προβλήματα εξαρτάται σε καθοριστικό βαθμό από τη χρήση αλγορίθμων και δομών δεδομένων.
- Κάθε πρόγραμμα βασίζεται σε αλγορίθμους και δομές δεδομένων.
- Ο σχεδιασμός των δομών δεδομένων αποτελεί καθοριστική απόφαση στον σχεδιασμό ενός προγράμματος.
- Αφού έχουμε επιλέξει τις δομές που θα χρησιμοποιήσουμε, ο σχεδιασμός ή η επιλογή των αλγορίθμων επίλυσης του προβλήματος, και ο προγραμματισμός του σε κάποια γλώσσα προγραμματισμού, είναι σχετικά εύκολες υποθέσεις αν γνωρίζουμε προγραμματισμό.
- Συνήθως, οι δομές δεδομένων και οι αλγόριθμοι που χρησιμοποιούμε είναι γνωστά και μελετημένα.
- Άρα, πρέπει να γνωρίζετε δομές δεδομένων και αλγορίθμους και τα χαρακτηριστικά τους, για να μπορείτε να επιλέγετε τα κατάλληλα για την επίλυση υπολογιστικών προβλημάτων.

M. Δικαιάκος, ΕΠΛ133

118

## Υπολογιστική Σκέψη (computational thinking)

- Ο τρόπος με τον οποίο οι επιστήμονες της Πληροφορικής χρησιμοποιούν:
  - **αποδόμηση** (decomposition)
  - **αναδρομή** (recursion)
  - **αλγορίθμους** (algorithms) και **δομές δεδομένων** για να αντιμετωπίσουν δύσκολα προβλήματα
- Η Υπολογιστική Σκέψη συγκεράζει όλα τα πλεονεκτήματα των υπολογιστικών μοντέλων, μεθόδων, εργαλείων και αφαιρέσεων που χρησιμοποιούμε.

M. Δικαιάκος, ΕΠΛ133

119

"World does not need software.  
It needs fully functional, easy to  
use systems."

Ragmar Saksing

M. Δικαιάκος, ΕΠΛ133

120



University of Cyprus  
Department of Computer Science

- **Κύκλος ζωής προγράμματος Java:** από τη σύνταξη στην εκτέλεση.
- Βασικές έννοιες: **Τύποι δεδομένων, Κλάσεις, Αντικείμενα.**
- Παρατήρηση: Η ενσωμάτωση της προγραμματιστικής επίλυσης ενός προβλήματος μέσα στη μέθοδο main **δεν αποτελεί Α/Σ προγραμματισμό.**

M. Δικαιάκος, ΕΠΛ133

## Προηγούμενα



University of Cyprus  
Department of Computer Science

- Κύκλος ζωής προγράμματος Java: από τη σύνταξη στην εκτέλεση.
- Βασικές έννοιες: Τύποι δεδομένων, Κλάσεις, Αντικείμενα.
- **Κλάση:** Ιδεότυπος / τύπος δεδομένων / δεδομένα + λειτουργίες.

M. Δικαιάκος, ΕΠΛ133

## Περίγραμμα



University of Cyprus  
Department of Computer Science

- Κλάσεις και Αντικείμενα
- Επίλυση Υπολογιστικών Προβλημάτων
- Παραδείγματα και Κατηγορίες Υπολογιστικών Προβλημάτων:
  - ▶ **Προβλήματα Βελτιστοποίησης**
  - ▶ Προβλήματα Αναπαράστασης Χώρου Καταστάσεων
- «Καλώς ορισμένα» προβλήματα

M. D. Dikaiakos

## ΠΡΟΒΛΗΜΑ:

### Εύρεση βέλτιστης διαδρομής

M. Δικαιάκος, ΕΠΛ133

124

### Επίλυση Προβλημάτων (Foody Express)

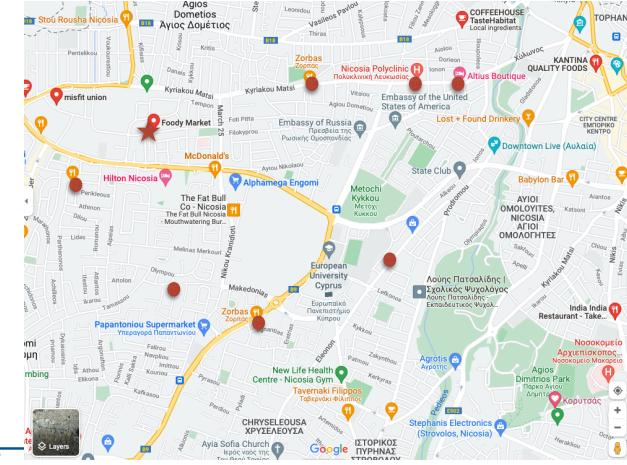
- Ορισμός: Βρες τον καλύτερο τρόπο για να παραδώσεις αντικείμενα στα 7 σημεία του χάρτη, ξεκινώντας και επιστρέφοντας στο αστέρι.
- Ανάλυση:
  - Ελάχιστη απόσταση, συντομότερη απόσταση ή χαμηλότερο κόστος;
  - Διαθέσιμα μέσα
  - Βάρος, όγκος μεταφερόμενων αντικειμένων

M. Δικαιάκος, ΕΠΛ133

126

### Επίλυση Προβλημάτων (Foody Express)

- Ορισμός προβλήματος: Βρες τον καλύτερο τρόπο για να παραδώσεις αντικείμενα στα 7 σημεία του χάρτη, ξεκινώντας και επιστρέφοντας στο αστέρι.

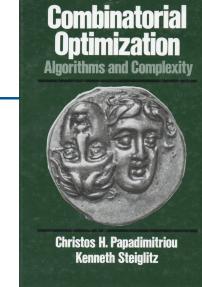


M. Δικαιάκος, ΕΠΛ133

125

### Επίλυση Προβλημάτος Εύρεσης Βέλτιστης Διαδρομής

- Αρχική Επινόηση Λύσης και Σχεδιασμός
  - Πρόβλημα **βελτιστοποίησης**
  - Λάβε υπόψη όλους τους δυνατούς δρόμους και όλα τα μέσα μεταφοράς
  - Βρες τον τρόπο υπολογισμού του **κόστους** μιας πιθανής διαδρομής
  - ...



M. Δικαιάκος, ΕΠΛ133

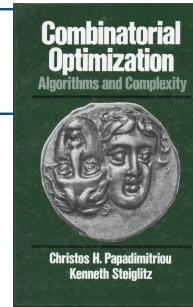
127

## Επίλυση Προβλημάτος Εύρεσης Βέλτιστης Διαδρομής

- Αρχική Επινόηση Λύσης και Σχεδιασμός
  - Πρόβλημα βελτιστοποίησης
  - Λάβε υπόψη όλους τους δυνατούς δρόμους και όλα τα μέσα μεταφοράς
  - Βρες τον τρόπο υπολογισμού του κόστους μιας πιθανής διαδρομής
  - ...
- Βελτιωμένος Σχεδιασμός
  - Ανάλυση όλων των πιθανών διαδρομών και μέσων μεταφοράς
  - Επιλογή της διαδρομής που πληροί τα κριτήρια μας (π.χ. συντομότερη και φθηνότερη)

M. Δικαιάκος, ΕΠΛ133

128

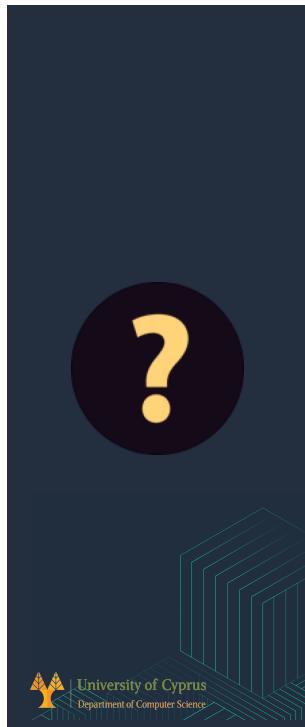


## Επίλυση Προβλημάτος

- Έλεγχος και αξιολόγηση:
  - Έλεγξε την ορθότητα της λύσης
  - Έλεγξε τα υπολογιζόμενα αποτελέσματα έναντι πραγματικών μετρήσεων
- Τεκμηρίωση λύσης
  - Υποβολή στον πελάτη περιγραφής της λύσης που έχουμε επιλέξει
- Συντήρηση λύσης
  - Συλλογή σχολίων από τον πελάτη για το ταξίδι του
  - Αναδιαμόρφωση της λύσης αν ο πελάτης δεν ήταν ικανοποιημένος από την εμπειρία

M. Δικαιάκος, ΕΠΛ133

129



M. D. Dikaiakos

## Περίγραμμα



- Επίλυση Υπολογιστικών Προβλημάτων
- Παραδείγματα και Κατηγορίες Υπολογιστικών Προβλημάτων:
  - ▶ Προβλήματα Βελτιστοποίησης
  - ▶ Προβλήματα Αναπαράστασης Χώρου Καταστάσεων
- «Καλώς ορισμένα» προβλήματα

## Αναπαράσταση χώρου καταστάσεων *state space representation*

- Τι είναι; Μια δημοφιλής προσέγγιση διατύπωσης και επίλυσης υπολογιστικών προβλημάτων
- Το πρόβλημα μοντελοποιείται σαν **χώρος καταστάσεων** (*state space*):
  - Σύνολο όλων των καταστάσεων στις οποίες μπορεί να «βρεθεί» ο αλγόριθμος επίλυσης κατά τη διάρκεια αναζήτησης λύσης
  - **Αρχικές καταστάσεις**: οι καταστάσεις από τις οποίες ξεκινάμε για να βρούμε λύση στο πρόβλημα
  - **Τελικές καταστάσεις**: καταστάσεις στις οποίες καταλήγουμε και συμπεριλαμβάνουν τις ενδεχόμενες λύσεις του προβλήματος
- Δύο καταστάσεις είναι συνδεδεμένες μεταξύ τους αν υπάρχει ένα **έγκυρο αλγορίθμικό βήμα** που μπορεί να **μετασχηματίσει** τη μια κατάσταση στην άλλη

M. Δικαιάκος, ΕΠΛ133

132

## ΠΡΟΒΛΗΜΑ:

### Τρίλιζα

M. Δικαιάκος, ΕΠΛ133

134

## Μετάπτωση κατάστασης

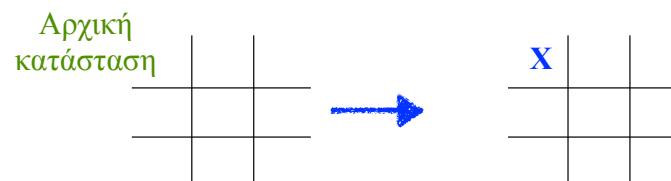
- Ένα «συμβάν» ενεργοποιεί την μετάβαση από τη μια κατάσταση στην άλλη
- Π.χ. ο χρήστης πατάει το κουμπί στο iPad και αυτό μεταπίπτει από κατάσταση “Sleep” σε “Operate” και πίσω.



M. Δικαιάκος, ΕΠΛ133

133

## Παράδειγμα: Τρίλιζα



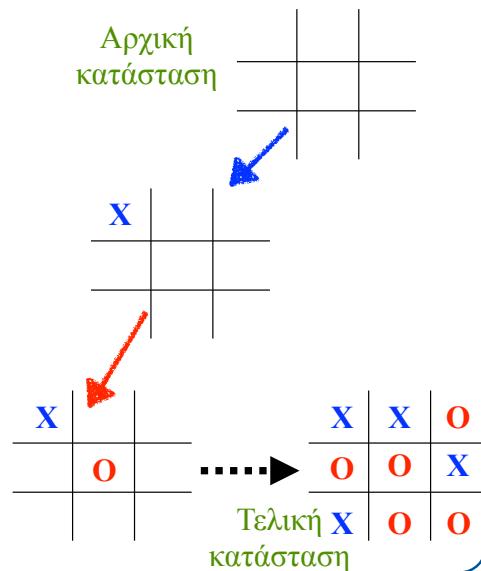
- Πώς κωδικοποιούμε τις καταστάσεις του προβλήματος;
- Μπορείτε να υπολογίσετε σε **πόσες δυνατές καταστάσεις** μπορεί να βρεθεί το παιχνίδι;
  - 9!
  - $2^9$
  - Κάτι άλλο;

M. Δικαιάκος, ΕΠΛ133

135

## Παράδειγμα: Τρίλιζα

- Κάθε κατάσταση αναπαριστά μια δυνατή τοποθέτηση του X και του O στην τρίλιζα.
- Το διάγραμμα των καταστάσεων αναπαριστά όλες τις δυνατές κινήσεις που μπορούν να γίνουν με βάση τους κανόνες του παιχνιδιού.



M. Δικαιάκος, ΕΠΛ133

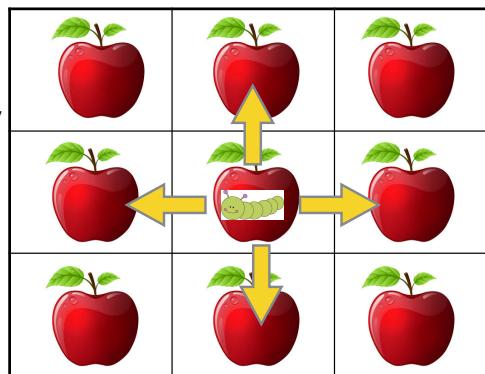
136

## ΠΡΟΒΛΗΜΑ:

### Τετράγωνα των μήλων

#### Πρόβλημα απόφασης: Τα τετράγωνα των μήλων

- **Πρόβλημα:** Ξεκινώντας από το κεντρικό τετράγωνο, μπορεί το σκουλήκι να φάει όλα τα μήλα;
- Κανόνες:
  - Το σκουλήκι μπορεί να μετακινηθεί μεταξύ τετραγώνων που έχουν κοινή ακμή, διασχίζοντας την ακμή.
  - Το σκουλήκι μετακινείται σε ένα τετράγωνο, μόνο αν υπάρχει αφάγωτο μήλο.
  - Το σκουλήκι είναι λαίμαργο: αν βρεθεί σε τετράγωνο με μήλο, θα το φάει.



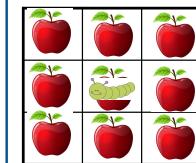
M. Δικαιάκος, ΕΠΛ133

138

M. Δικαιάκος, ΕΠΛ133

137

#### ΕΙΣΟΔΟΣ (input)



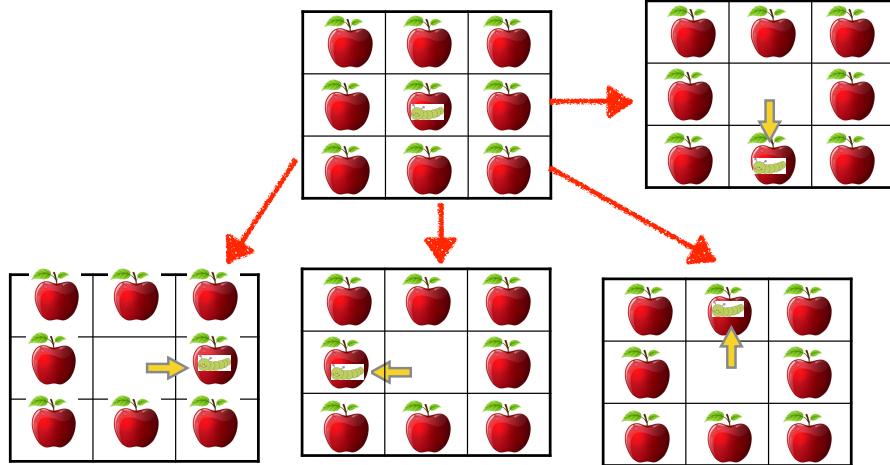
```
/** * Created by mdd on 15/01/2017. */  
public class LeapYear {  
    int year = Integer.parseInt(args[0]);  
    // divisible by 4  
    boolean isLeapYear = (year % 4 == 0);  
    // divisible by 4 and not 100  
    isLeapYear = isLeapYear && (year % 100 != 0);  
    // divisible by 4 and not 100 unless divisible by 400  
    isLeapYear = isLeapYear || (year % 400 == 0);  
    System.out.println(isLeapYear);  
}
```

ΕΞΟΔΟΣ (output)  
ΝΑΙ  
ή  
ΟΧΙ

M. Δικαιάκος, ΕΠΛ133

139

## Αναπαράσταση καταστάσεων



Μ. Δικαιάκος, ΕΠΛ133

140

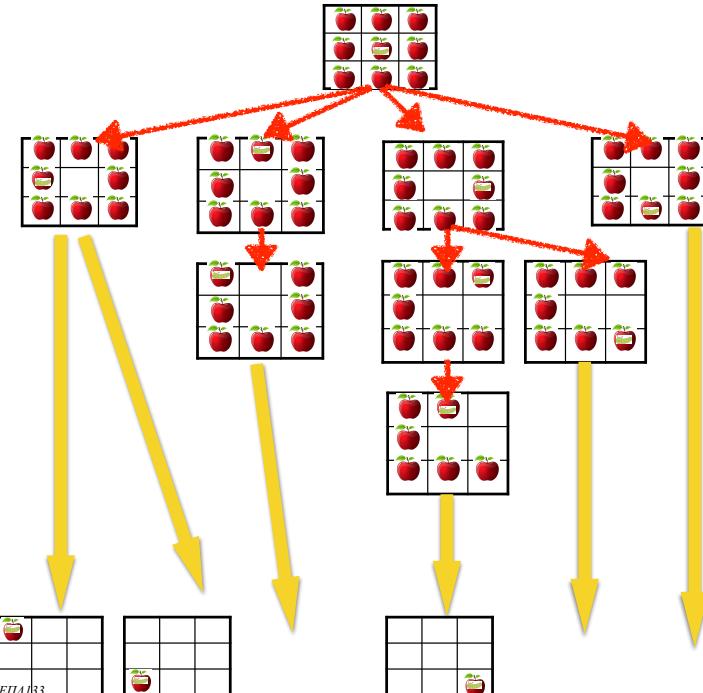
Πόσες διαφορετικές διαδρομές μπορεί να ακολουθήσει το σκουλήκι για να φάει όλα τα μήλα;

**00:88**

- Καμία - δεν υπάρχει λύση
- 1
- 8
- 32

ΕΠΛ233

142

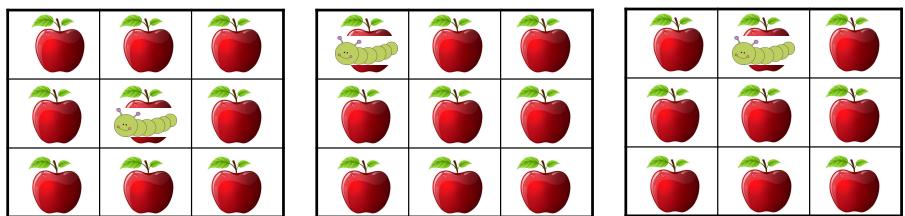


Μ. Δικαιάκος, ΕΠΛ133

141

## Τα τετράγωνα των μήλων II

- Πόσες και ποιες διαδρομές μπορεί να ακολουθήσει το σκουλήκι, ξεκινώντας από ένα οποιοδήποτε τετράγωνο, ώστε να φάει όλα τα μήλα;

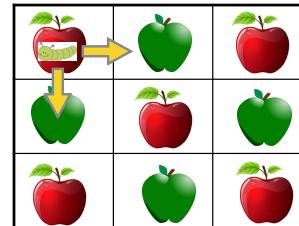
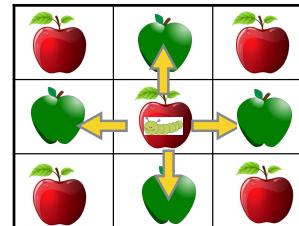


Μ. Δικαιάκος, ΕΠΛ133

143

## Τα τετράγωνα των πολύχρωμων μήλων

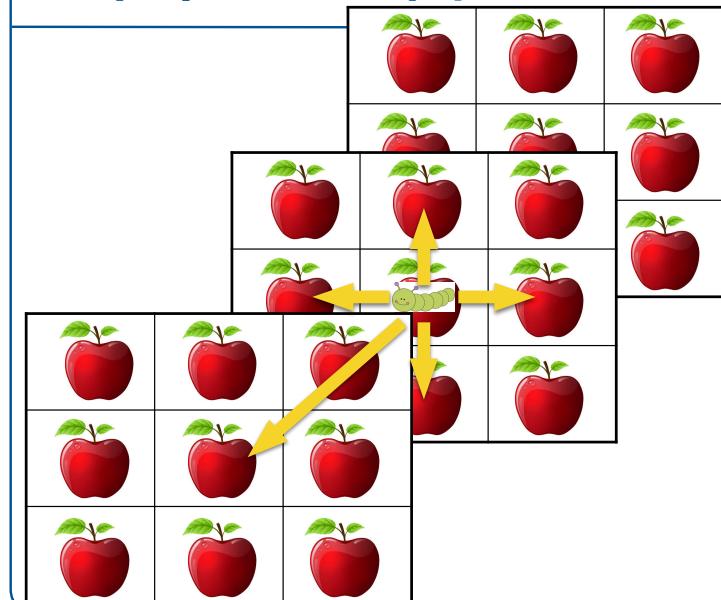
- Ξεκινώντας από το κεντρικό τετράγωνο, το σκουλήκι μπορεί να φάει όλα τα μήλα;
- Κανόνες:
  - Το σκουλήκι μπορεί να μετακινηθεί μεταξύ τετραγώνων που έχουν κοινή ακμή αλλά **διαφορετικό χρώμα μήλου**.
  - Το σκουλήκι επισκέπτεται ένα τετράγωνο μια μόνο φορά.
- Πόσες λύσεις (διαδρομές) έχουμε αν το σκουλήκι:
  - Ξεκινήσει από το κέντρο;
  - Ξεκινήσει από άκρη με πράσινο ή με κόκκινο μήλο;



Μ. Δικαιάκος, ΕΠΛ133

144

## Τετράγωνα των μήλων σε 3 διαστάσεις



Μ. Δικαιάκος, ΕΠΛ133

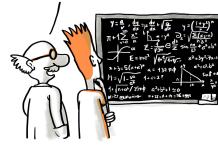
145

## Περίγραμμα



- Επίλυση Υπολογιστικών Προβλημάτων
- Παραδείγματα και Κατηγορίες Υπολογιστικών Προβλημάτων
- «Καλώς ορισμένα» προβλήματα

## Ανασκόπηση



### ■ Τι είδαμε Μέχρι Τώρα;

- Εξετάσαμε απλά και σύνθετα προβλήματα και συζητήσαμε πώς μπορούμε να τα αναλύσουμε για να τους δώσουμε προγραμματιστική επίλυση
- Είδαμε παραδείγματα προβλημάτων με διάφορα χαρακτηριστικά και ιδιότητες.

### ■ Τι μάθαμε;

- Την ύπαρξη προβλημάτων **βελτιστοποίησης** - στα οποία αναζητούμε μια λύση με βέλτιστα χαρακτηριστικά - και προβλημάτων **απόφασης** - στα οποία η λύση τους είναι απάντηση ΝΑΙ/ΟΧΙ ή ΣΩΣΤΟ/ΛΑΘΟΣ.
- Η επίλυση ενός προβλήματος προϋποθέτει την κατάλληλη διατύπωση-θεμελίωση του, την αποδόμηση του σε (ευκολότερα) υποπροβλήματα (**διαίρει-και-βασίλευε**), την ανεύρεση (ή τον σχεδιασμό) καταλλήλων δομών δεδομένων και αλγορίθμων, προγραμματισμό σε μια Γ/Π, τεκμηρίωση, αποσφαλμάτωση, αξιολόγηση, βελτίωση κλπ.
- Η θεμελίωση απλών προβλημάτων μπορεί να καταλήξει σε υλοποίηση μέσω μιας μαθηματικής **συνάρτησης** ή μιας σειράς **λογικών ελέγχων**. Σε ποι σύνθετα προβλήματα η θεμελίωση μπορεί να προϋποθέτει την αποτύπωση διαδοχικών καταστάσεων του προβλήματος σε κάποια κατάλληλη δομή δεδομένων.

### ■ Τι ακολουθεί Μετά;

- «Καλώς ορισμένα» προβλήματα

M. Δικαιάκος, ΕΠΛ133

148

## «Καλώς ορισμένα» Προβλήματα

- Ποιό είναι το αποτέλεσμα της πράξης; **“ $1 \times 2 + 3$ ”**
- Ποιό είναι το αποτέλεσμα της πράξης; **“ $1 + 2 \times 3$ ”**
- **Foody Express problem:** Ποιά είναι η **βέλτιστη διαδρομή** για την παράδοση δεμάτων σε 7 σημεία στον ακόλουθο χάρτη;



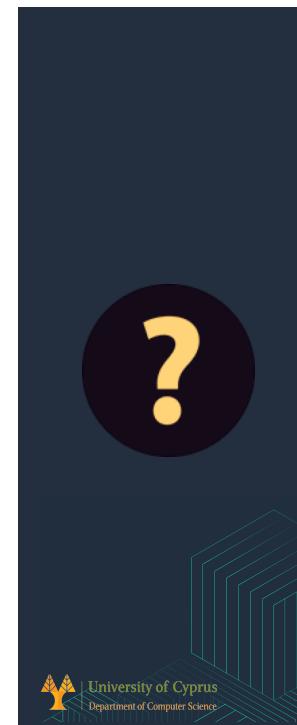
150

## «Καλώς ορισμένα» Προβλήματα (well-defined problems)

- Μπορεί να διατυπωθεί ένας **σαφής** και **αυστηρά διατυπωμένος** (τυπικός-formal) ορισμός του προβλήματος, από την περιγραφή του σε φυσική γλώσσα;
  - Ενδεχομένως με την υιοθέτηση διασαφηνιστικών/ αφαιρετικών υποθέσεων που δεν αλλοιώνουν την υφή του προβλήματος
- Υπάρχει δυνατότητα εύρεσης λύσης που είναι αποδεδειγμένα σωστή (ή μαθηματική **απόδειξη** ότι δεν μπορεί να βρεθεί λύση)
- Η λύση μπορεί να βρεθεί σε πεπερασμένο αριθμό αλγορίθμικών βημάτων;

M. Δικαιάκος, ΕΠΛ133

149



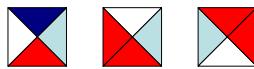
M. Δικαιάκος, ΕΠΛ133



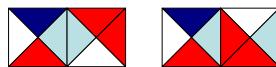
M. D. Dikaiakos

## Πλακόστρωτο μονοπάτι (tiling problem)

- Σας δίνεται ένας πεπερασμένο σύνολο από διαφορετικά έγχρωμα πλακάκια.



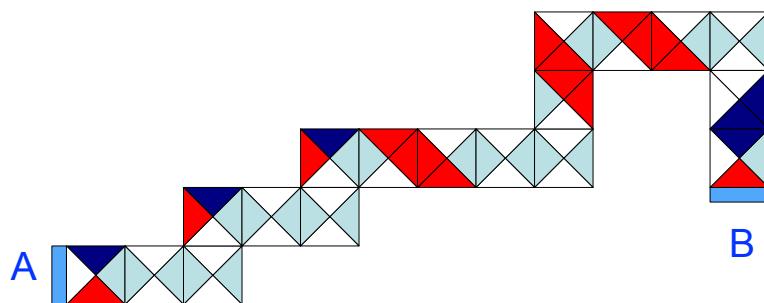
- Επιτρέπεται να τοποθετήσετε δύο πλακάκια δίπλα-δίπλα μόνο αν οι συνορεύουσες ακμές τους έχουν το ίδιο χρώμα.



**OK**

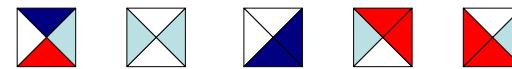
**OXI**

- Δεν επιτρέπεται να περιστρέψετε τα πλακάκια.
- Διαθέτετε απεριόριστο αριθμό από κάθε είδος.



## Πλακόστρωτο μονοπάτι

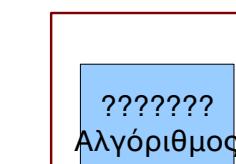
- Σας δίνεται ένας αριθμός από πλακάκια διαφορετικών χρωμάτων και δύο σημεία στο επίπεδο Α και Β.



- Αναζητείτε έναν αλγόριθμο που να μπορεί να λύσει το ακόλουθο πρόβλημα:

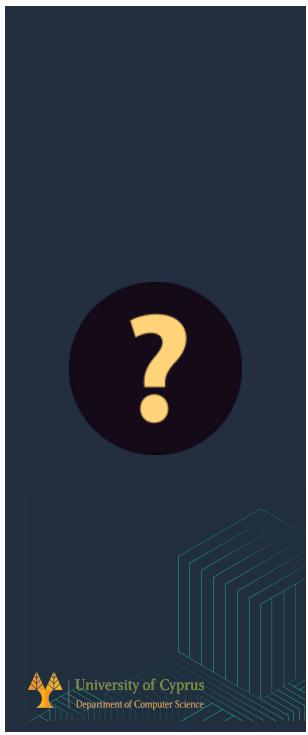
Μπορούμε να συνδέσουμε τα σημεία Α και Β με ένα πλακόστρωτο μονοπάτι, όταν το φιδάκι επιτρέπεται να “κινηθεί” οπουδήποτε στο επίπεδο: ΝΑΙ ή ΟΧΙ ;

## Το πλακόστρωτο μονοπάτι ως υπολογιστικό πρόβλημα

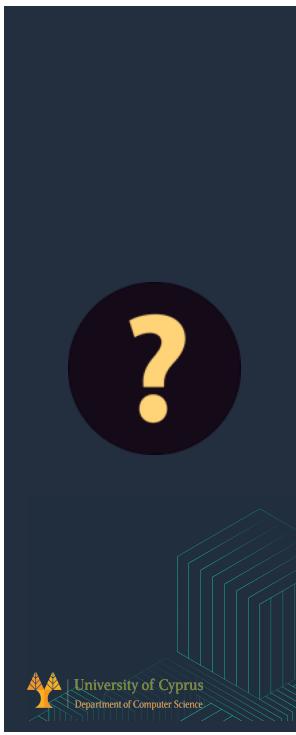


A    B

ΝΑΙ/ΟΧΙ



M. D. Dikaiakos



M. D. Dikaiakos

## Πότε υπάρχει λύση;

Δεδομένου ενός αριθμού από πλακάκια και δύο σημείων στο επίπεδο A και B, μπορούμε να συνδέσουμε τα σημεία A και B με ένα “φιδάκι” από πλακάκια, όταν το φιδάκι επιτρέπεται να “κινηθεί” οπουδήποτε στο επίπεδο: NAI ή OXI ;

- Αποδεικνύεται ότι υπάρχει λύση (αλγόριθμος) όταν το φιδάκι μπορεί να επεκταθεί ελεύθερα σε ολόκληρο το επίπεδο:
- Αν όμως χωρίσουμε το επίπεδο στα δύο και περιορίσουμε την κίνηση του φιδιού στο μισό επίπεδο τότε αποδεικνύεται ότι δεν υπάρχει αλγόριθμος που να μπορεί να μας πει NAI ή OXI!!