ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

ΑΝΟΙΚΤΑ ακαδημαϊκά μαθήματα ΠΠ

Τεχνολογίες Υλοποίησης Αλγορίθμων

Χρήστος Ζαρολιάγκης

Καθηγητής
Τμήμα Μηχ/κων Η/Υ & Πληροφορικής
Πανεπιστήμιο Πατρών
email: zaro@ceid.upatras.gr

Ενότητα 3

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

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 Πανεπιστήμιο Πατρών»** 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.

Περιεχόμενα

- Δοκιμή Προγραμμάτων (Program Testing)
- Έλεγχος Ορθότητας Προγραμμάτων (Program Correctness Checking)

Δοκιμή Προγράμματος

- εκτέλεση του προγράμματος σε υπολογιστή χρησιμοποιώντας *δεδομένα δοκιμής (test data)*
- Σύγκριση μεταξύ πραγματικής και αναμενόμενης (θεωρητικής) συμπεριφοράς του

Δοκιμή Προγράμματος – Στόχος

- Δοκιμή (testing) $\neq$ Έλεγχος ορθότητας (correctness checking)

  αριθμός των διαφορετικών εισόδων μπορεί να είναι γενικά πολύ μεγάλος

  $\Rightarrow$

  η δοκιμή περιορίζεται συχνά σε ένα μικρό υποσύνολο όλων των δυνατών εισόδων – *σύνολο δοκιμής (test set)*

  $\Rightarrow$

  σύνολο δοκιμής δεν μπορεί να μας δώσει ορθότητα προγράμματος

- **Στόχος Δοκιμής:** όχι η εξασφάλιση ορθότητας, αλλά ο εντοπισμός των λαθών

  ``*Program testing can be used to show the presence of bugs, but never to show their absence!*'' (Dijkstra, 1972)

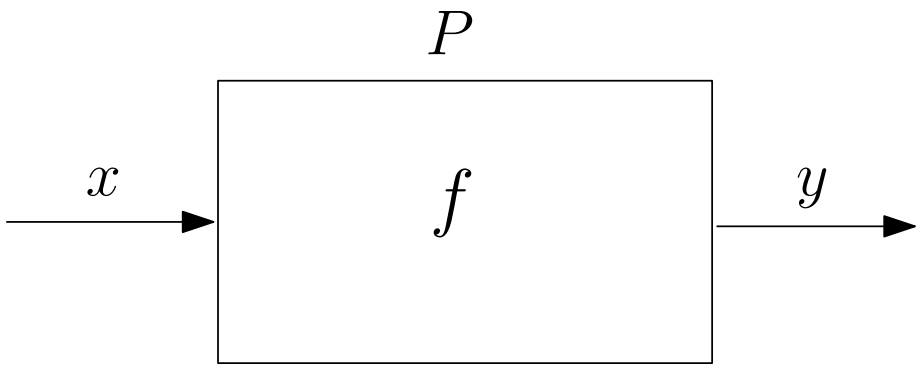
- **Μέθοδοι Black-Box**
  - Λαμβάνουν υπόψιν τη λειτουργία του αλγορίθμου και όχι του κώδικα
  - Διαχωρισμός δεδομένων εισόδου σε διαφορετικές ομάδες που η κάθε μία αντιπροσωπεύει μια ποιοτικά διαφορετική συμπεριφορά

- **Μέθοδοι White-Box**
  - Σχεδιασμός δεδομένων δοκιμής μετά από λεπτομερή εξέταση του κώδικα, προκειμένου να καλύψουν όλες τις εντολές του προγράμματος και όλες τις πιθανές «διαδρομές» εκτέλεσης

Έλεγχος Ορθότητας Προγραμμάτων

Διαδικασία που μας βοηθά να αυξήσουμε σημαντικά
την εμπιστοσύνη μας σε μια υλοποίηση

- Έστω P ένα πρόγραμμα το οποίο υπολογίζει μια συνάρτηση f



- Πώς μπορούμε να βεβαιωθούμε ότι το P, με είσοδο x, όντως υπολογίζει $y = f(x)$;

Έλεγχος Ορθότητας Προγραμμάτων – Αρνητικά Παραδείγματα

- Ο αλγόριθμος ελέγχου γραφήματος ως προς το αν είναι επίπεδο ήταν εσφαλμένος στην LEDA 2.0
- Το σύστημα (CAD) Rhino3d αποτυγχάνει να υπολογίσει σωστά την τομή δύο κυλίνδρων και δύο σφαιρών
- Ο επιλυτής γραμμικών προγραμμάτων CPLEX αποτυγχάνει στο πρόβλημα δοκιμής επιδόσεων (benchmark problem) etamacro

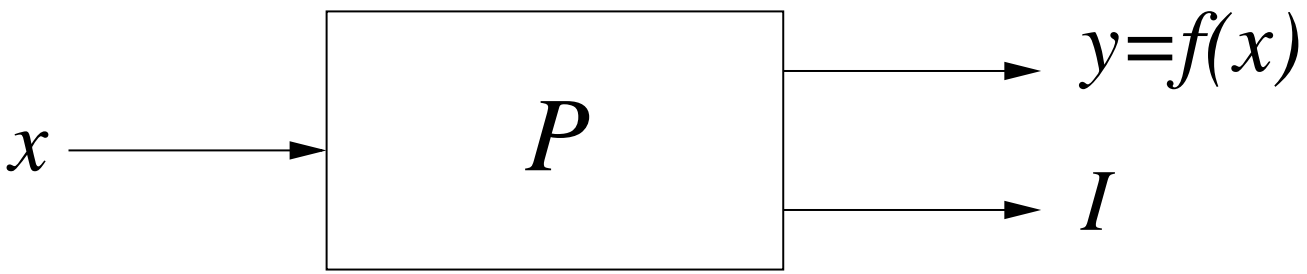
Έλεγχος Ορθότητας Προγραμμάτων – Δόγμα

Τα προγράμματα πρέπει να δικαιολογούν (αποδεικνύουν) την έξοδό τους με έναν τρόπο που να μπορεί εύκολα να επαληθευθεί από κάποιον χρήστη

Προγράμματα Πιστοποίησης (Certifying Programs)

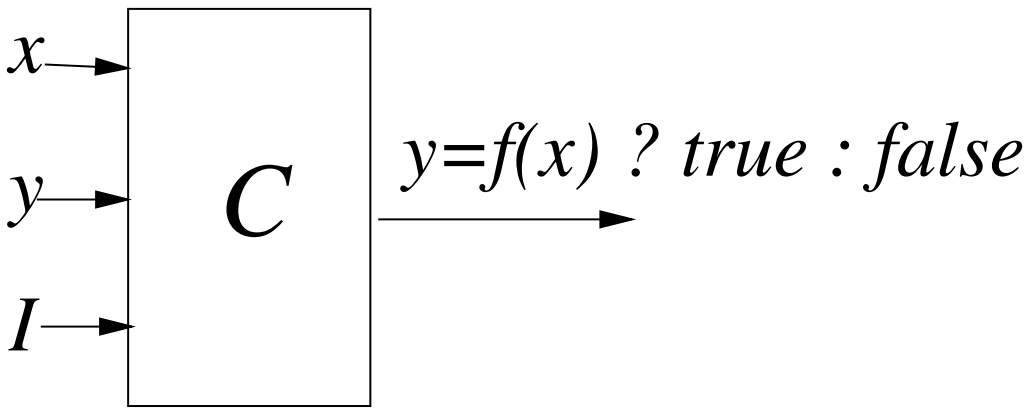
- Ένα πρόγραμμα P καλείται **πρόγραμμα πιστοποίησης** (ή *ελέγξιμο*) αν $\forall$ είσοδο x επιστρέφει
  - y, την φερόμενη τιμή της $f(x)$, και
  - ένα **πιστοποιητικό** (ή *πληροφορία επιβεβαίωσης*) I

  το οποίο καθιστά **εύκολα επαληθεύσιμο** ότι όντως $y = f(x)$



Εύκολα επαληθεύσιμο

- $\exists$ απλό πρόγραμμα C, ο **ελεγκτής**, το οποίο δεδομένων των x, y, και I ελέγχει αν όντως $y = f(x)$
  Το C πρέπει να είναι τόσο απλό που η ορθότητά του να είναι προφανής



- Ο χρόνος εκτέλεσης (πραγματικός ή ασυμπτωτικός) του C με είσοδο x, y και I δεν πρέπει να είναι μεγαλύτερος από εκείνον του P με είσοδο x

Έλεγχος Ορθότητας Προγραμμάτων – Παραδείγματα

- Επίλυση γραμμικών συστημάτων
- Εύρεση συντομότερων αποστάσεων
- Έλεγχος επιπεδότητας γραφημάτων
- Έλεχγος διμερότητας γραφημάτων

Έλεγχος Ορθότητας Προγραμμάτων – Επίλυση γραμμικών συστημάτων

- Θεωρήστε ένα πρόγραμμα P_1 το οποίο παίρνει σαν είσοδο ένα μητρώο $A_{m\times n}$ και ένα διάνυσμα $b_{m\times 1}$ και εξετάζει άν το γραμμικό σύστημα $A \cdot x = b$ έχει λύση ή όχι, επιστρέφοντας μια λογική τιμή true ή false

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & a_{2,3} & \dots & a_{2,n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{m,1} & a_{m,2} & a_{m,3} & \dots & a_{m,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_m \end{bmatrix}$$

- Είναι το P_1 ένα πρόγραμμα πιστοποίησης (ελέγξιμο) ;
- Αν όχι, τότε πώς μπορεί να γίνει ;

Έλεγχος Ορθότητας Προγραμμάτων – Επίλυση γραμμικών συστημάτων

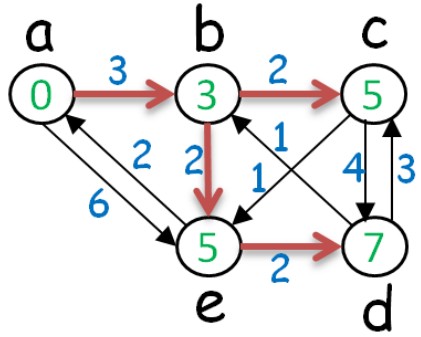
Το P_1, ως έχει, ***δεν*** είναι πρόγραμμα πιστοποίησης
Για να γίνει, πρέπει να επεκταθεί η διασύνδεσή του

- Με είσοδο A και b ένα **πρόγραμμα πιστοποίησης** P_1' επιστρέφει
  - είτε «το σύστημα έχει λύση» και ένα διάνυσμα x τέτοιο ώστε $A \cdot x = b$
  - είτε «το σύστημα δεν έχει λύση» και ένα διάνυσμα c τέτοιο ώστε $c^T \cdot A = 0$ και $c^T \cdot b \neq 0$

- Τώρα το P_1' είναι εύκολα επαληθεύσιμο:
  απαιτούνται το πολύ δύο (απλοί και εύκολα ελέγξιμοι) πολλαπλασιασμοί μητρώου με διάνυσμα και διάνυσμα με διάνυσμα

Έλεγχος Ορθότητας Προγραμμάτων – Εύρεση συντομότερων αποστάσεων

- Θεωρήστε ένα πρόγραμμα P_2 που δέχεται ως είσοδο ένα κατευθυνόμενο γράφημα $G = (V, E)$, με κόστη στις πλευρές του $wt : E \to \mathbb{R}$ και με μια κορυφή αφετηρίας $s \in V$, και το οποίο υπολογίζει (μόνο) τις συντομότερες αποστάσεις $d(v), \forall v \in V$, από την s

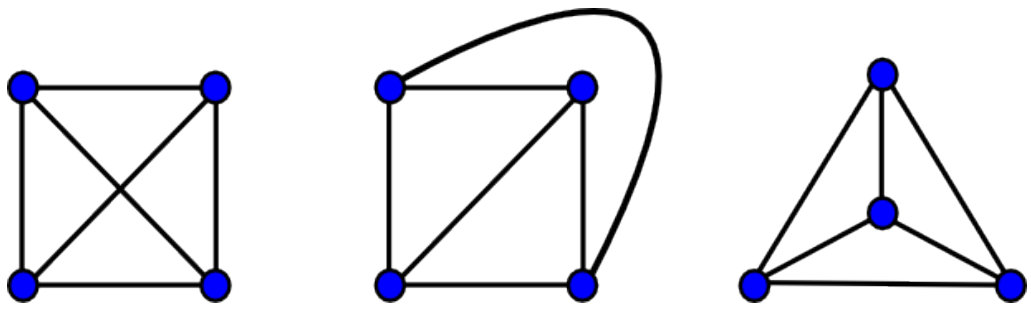


- Είναι το P_2 ένα πρόγραμμα πιστοποίησης (ελέγξιμο) ;
- Αν όχι, τότε πώς μπορεί να γίνει ;
- Το P_2, ως έχει, είναι **όντως πρόγραμμα πιστοποίησης**

  $\forall (u, v) \in E$: $d(v) \leq d(u) + wt(u, v)$,
  με την ισότητα να ισχύει για τουλάχιστον $n - 1$ ακμές ($n = |V|$)

Έλεγχος Ορθότητας Προγραμμάτων – Έλεγχος επιπεδότητας γραφημάτων

- Θεωρήστε ένα πρόγραμμα P_3 το οποίο δέχεται ως είσοδο ένα γράφημα G και εξετάζει αν το G είναι επίπεδο ή όχι, επιστρέφοντας μια λογική τιμή true ή false

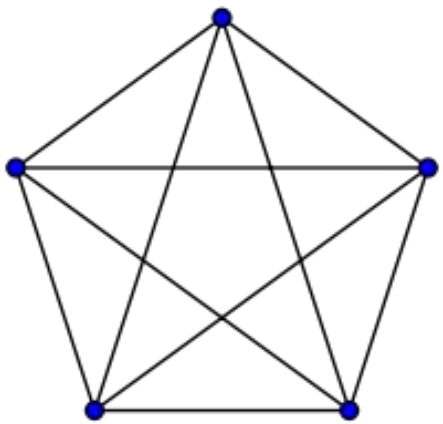


Εικόνα 1

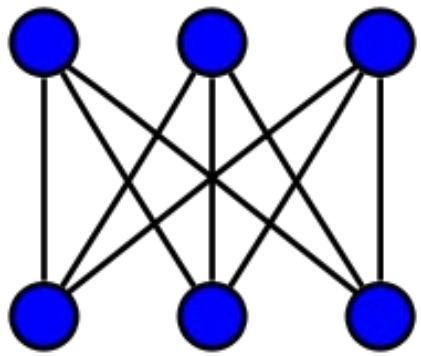
- Είναι το P_3 ένα πρόγραμμα πιστοποίησης (ελέγξιμο) ;
- Αν όχι, τότε πώς μπορεί να γίνει ;

Το P_3, ως έχει, ***δεν*** είναι πρόγραμμα πιστοποίησης
Για να γίνει, πρέπει να επεκταθεί η διασύνδεσή του

- Με είσοδο G το **πρόγραμμα πιστοποίησης** P_3' επιστρέφει
  - είτε «επίπεδο» μαζί με μια επίπεδη απεικόνιση του G
  - είτε «μη-επίπεδο» μαζί με ένα υπογράφημα του G ομοιομορφικό με το K_5 ή το $K_{3,3}$ (υπογραφήματα Kuratowski)



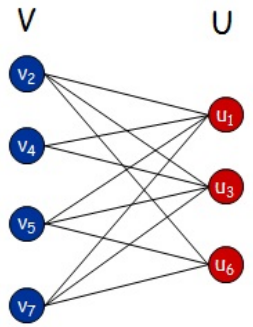
Εικόνα 2: K_5



Εικόνα 3: $K_{3,3}$

Έλεγχος Ορθότητας Προγραμμάτων – Έλεγχος διμερότητας γραφημάτων

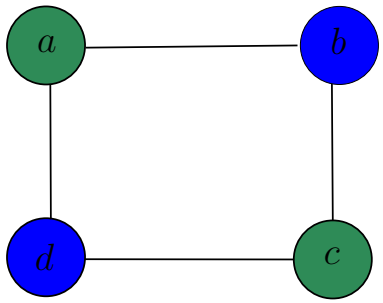
- Θεωρήστε ένα πρόγραμμα P_4 που δέχεται ως είσοδο ένα γράφημα G και εξετάζει αν το G είναι διμερές ή όχι, επιστρεφοντας μια λογική τιμή true ή false



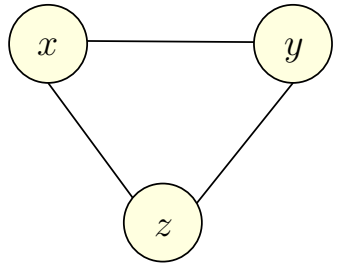
- Είναι το P_4 ένα πρόγραμμα πιστοποίησης (ελέγξιμο) ;
- Αν όχι, τότε πώς μπορεί να γίνει ;

Έλεγχος Ορθότητας Προγραμμάτων – Έλεγχος διμερότητας γραφημάτων

Το P_4, ως έχει, ***δεν*** είναι πρόγραμμα πιστοποίησης



2-χρωματισμός πιστοποιεί διμερότητα



περιττός κύκλος πιστοποιεί μη-διμερότητα

Πρόγραμμα (αλγόριθμος) πιστοποίησης P_4':

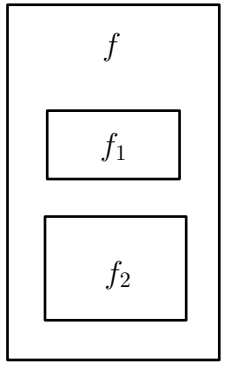
- Κατασκευή γεννητικού δένδρου και χρήση του για χρωματισμό κορυφών του G με χρώματα πράσινο και μπλε
- για κάθε μη-δενδρική ακμή, έλεγχος αν τα άκρα της έχουν διαφορετικά χρώματα
  - Αν ναι, τότε το G είναι διμερές (και ο χρωματισμός το αποδεικνύει)
  - Αν όχι, τότε το G δεν είναι διμερές
    - Έστω $e = (x, y)$ μια μη-δενδρική ακμή με άκρα ίδιου χρώματος
    - Η δενδρική διαδρομή x-y έχει άρτιο μήκος (x & y έχουν ίδιο χρώμα)
    - Η ακμή e με την δενδρική διαδρομή x-y είναι ένας περιττός κύκλος

1. Η απάντηση του προγράμματος μπορεί να επαληθευθεί **για κάθε** ξεχωριστό στιγμιότυπο εισόδου $\neq$ επαλήθευση προγράμματος (program verification) η οποία παρέχει εγγύηση για όλα τα στιγμιότυπα εισόδου

2. Οι αλγόριθμοι πιστοποίησης είναι **αξιόπιστοι**
   - Είτε δίνουν τη σωστή απάντηση
   - Είτε παρέχουν ένδειξη ότι υπάρχει σφάλμα

3. **Υπολογισμός αυξημένης εμπιστοσύνης**
   - Ο χρήστης μπορεί να αυξήσει την ***εμπιστοσύνη*** του στην ορθότητα του προγράμματος καταβάλοντας πολύ μικρή διανοητική προσπάθεια
     $\nexists$ ανάγκη να καταλάβει το πρόγραμμα – αρκεί να καταλάβει την ιδιότητα του πιστοποιητικού και τον ελεγκτή ορθότητας
   - Ο υλοποιητής ενός προγράμματος μπορεί να δώσει μια ***πειστική ένδειξη*** της ορθότητάς του χωρίς να αποκαλύψει οποιαδήποτε λεπτομέρεια της υλοποίησης

4. Ο έλεγχος ορθότητας επιτρέπει να χρησιμοποιούμε ένα πιθανώς μη-ορθό πρόγραμμα σαν να ήταν ορθό $\Rightarrow$ πολύ **χρήσιμο κατά τη διαδικασία αποσφαλμάτωσης**



5. Ο έλεγχος ορθότητας **υποστηρίζει τη δοκιμή** προγράμματος (testing)

```
for (int n = 0; n < 100; n++)
  for (int m = 0; m < 100; m++)
  { random_graph(G,n,m);
    // random graph with n nodes and m edges
    list<edge> M = MAX_CARD_MATCHING(G,OSC);
    CHECK_MAX_CARD_MATCHING(G,M,OSC);
  }
```

6. Ένας ελεγκτής (πρόγραμμα ελέγχου ορθότητας) μπορεί να γραφεί αν υπάρχει αυστηρός ορισμός του προβλήματος που επιλύει το πρόγραμμα που ελέγχει

   Π.χ. αν ένας αλγόριθμος επίλυσης ενός προβλήματος γραφημάτων υποθέτει ότι δεν υπάρχουν μεμονωμένοι κόμβοι στο γράφημα, τότε η ίδια υπόθεση πρέπει να τηρηθεί και από τον ελεγκτή

Έλεγχος Ορθότητας Προγραμμάτων – Άλλα ζητήματα

- **Καθολικότητα:** Έχει κάθε πρόβλημα έναν αλγόριθμο πιστοποίησης ; Μπορεί κάθε πρόγραμμα να μετατραπεί σε πρόγραμμα πιστοποίησης ;
- Κάθε αιτιοκρατικό πρόγραμμα μπορεί να μετατραπεί σε πρόγραμμα πιστοποίησης (McConnell, Mehlhorn, Naeher, Schweitzer, 2011)
- **Τυπική Επαλήθευση (formal verification):** προσθέτει ένα επιπλέον επίπεδο εμπιστοσύνης

  Τυπικές αποδείξεις

  - είναι ορθές και πλήρεις
  - είναι μηχανικά ελεγχόμενες (από ένα πολύ απλό πρόγραμμα)
  - επιτρέπουν τη δημιουργία μεγάλων βιβλιοθηκών έμπιστων αλγορίθμων
  - επιτρέπουν στον χρήστη να καταβάλλει ακόμα μικρότερη διανοητική προσπάθεια

    εμπιστευθειτε απλώς τον ελεγκτή απόδειξης !

Σύνοψη

- Δοκιμή προγραμμάτων $\neq$ Έλεγχος Ορθότητας Προγραμμάτων
- Αλγόριθμοι/προγράμματα πιστοποίησης: ιδιαίτερα επωφελή ...
  - μπορούν να ελεγχθούν σε κάθε είσοδο
  - είναι αξιόπιστα
  - μπρούμε να τα εμπιστευθούμε χωρίς να γνωρίζουμε τον κώδικα
  - υποστηρίζουν τον υπολογισμό αυξημένης εμπιστοσύνης

  ... και παρέχουν έναν ***νέο τρόπο*** ανάπτυξης αλγορίθμων

Τέλος Ενότητας

Σημείωμα Ιστορικού Εκδόσεων Έργου

Το παρόν έργο αποτελεί την έκδοση **1.0**.

Σημείωμα Ιστορικού Εκδόσεων Έργου

Σημείωμα Ιστορικού Εκδόσεων Έργου

Πηγές εικόνων - Χρήση Έργων Τρίτων

Εικόνα 1:
http://www.boost.org/doc/libs/1_36_0/libs/graph/doc/figs/planar_plane_straight_line.png

Εικόνα 2:
https://commons.wikimedia.org/wiki/File:Complete_graph_K5.svg

Εικόνα 3:
https://commons.wikimedia.org/wiki/File:Biclique_K_3_3.svg

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