



ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ  
**ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΚΑΙ ΔΙΚΤΥΩΝ**  
ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ



Εργασία στο μάθημα “Αλγόριθμοι και Πολυπλοκότητα”  
Χρονοπρογραμματισμός εξετάσεων Πανεπιστημίου V1.0

Μπουραζάνης Γεώργιος | ΑΜ: 84

31/01/2021

# Περιεχόμενα

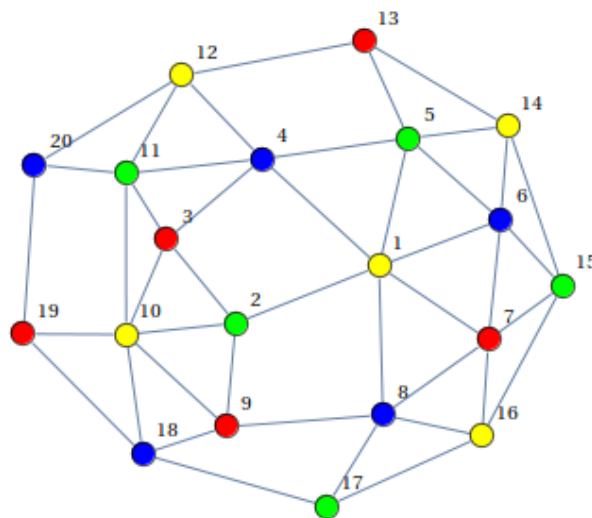
Περίληψη.....	3
1. Εισαγωγή (Προβλήματα NP-Complete) .....	4
2. Περιγραφή του Προβλήματος Χρωματισμού Γραφημάτων .....	5
3. Προσεγγίσεις Επίλυσης .....	7
3.1. Δεδομένα Προβλήματος (Toronto Datasets).....	7
3.2. Περιγραφή του Προγράμματος .....	8
3.2.1. Σύντομη Περιγραφή του Κώδικα .....	8
4. Επίλυση Προβλήματος.....	9
4.1. First Fit.....	9
4.2. Greedy Graph Coloring .....	9
5. Αποτελέσματα .....	10
6. Συμπεράσματα .....	11
Αναφορές.....	11

## Περίληψη

Ο χρονοπρογραμματισμός αποτελεί σημαντικό κομμάτι στην εκπαίδευση και κυρίως στο χρονοδιάγραμμα των εξετάσεων. Ο χρονοπρογραμματισμός εξετάσεων είναι μια ιδιαίτερα σημαντική λόγω της πολυπλοκότητας του προβλήματος προκειμένου να αποφύγουμε συγκρούσεις μεταξύ των μαθημάτων. Σε αυτή την εργασία θα δούμε την επίλυση του προβλήματος χρονοπρογραμματισμού εξετάσεων σε 13 προβλήματα από διάφορα εκπαιδευτικά ιδρύματα μέσω της χρήσης αλγορίθμων χρωματισμού γραφημάτων.

## 1. Εισαγωγή (Προβλήματα NP-Complete)

Το πρόβλημα του χρωματισμού γραφήματος είναι ένα NP-hard πρόβλημα συνδυαστικής βελτιστοποίησης. Αφορά την ανάθεση ενός χρώματος σε κάθε κορυφή ενός γραφήματος έτσι ώστε γειτονικές κορυφές να χρωματίζονται με διαφορετικό χρώμα (όπως στο ακόλουθο σχήμα), ενώ παράλληλα χρησιμοποιείται ο ελάχιστος αριθμός διαφορετικών χρωμάτων.



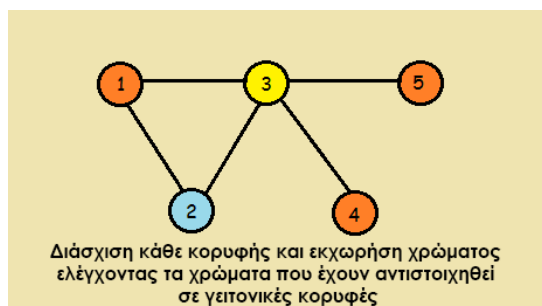
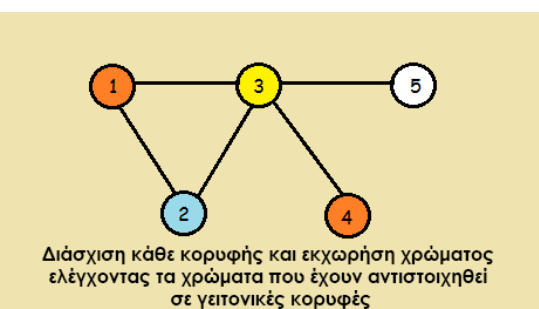
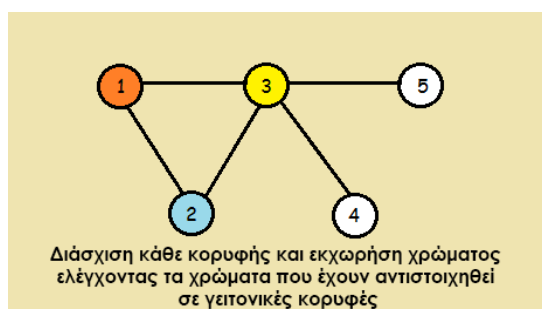
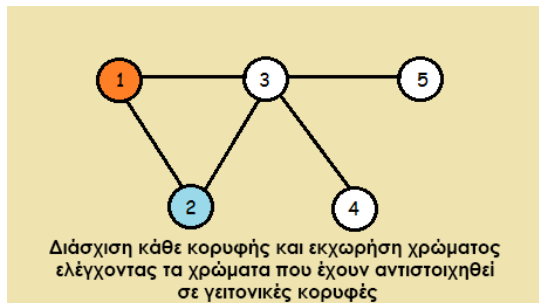
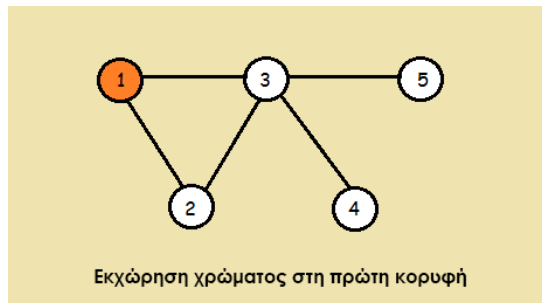
Στη συγκεκριμένη εργασία θέλουμε να παρουσιάσουμε μια λύση για έναν αριθμό προβλημάτων χρονοπρογραμματισμού. Στο δεύτερο κεφάλαιο θα περιγράψουμε το πρόβλημα χρωματισμού γραφημάτων και πιο συγκεκριμένα τα προβλήματα που αφορούν τα εκπαιδευτικά ιδρύματα. Στο τρίτο κεφάλαιο θα δούμε τη προσέγγιση επίλυσης του προβλήματος φορτώνοντας τα δεδομένα (Toronto Datasets) τα οποία θα χρησιμοποιηθούν για την επίλυση καθώς και την δομή με την οποία είναι αποθηκευμένα. Στο τέταρτο κεφάλαιο θα δούμε το τρόπο λειτουργίας της συνάρτησης καθώς και περιγραφή των κυριότερων συναρτήσεων που χρησιμοποιούνται μέσα στον κώδικα. Τέλος, θα περιγράψουμε τον τρόπο επίλυσης των δεδομένων με την χρήση του Greedy Coloring Algorithm και τα αποτελέσματα από την επίλυση των προβλημάτων καθώς επίσης θα γίνει και περιγραφή του First Fit αλγορίθμου.

## 2. Περιγραφή του Προβλήματος Χρωματισμού Γραφημάτων

Ο χρωματισμός γραφήματος είναι μια εκχώρηση ετικετών που παραδοσιακά ονομάζονται "graph colors" σε στοιχεία ενός γραφήματος που υπόκεινται σε ορισμένους περιορισμούς. Δηλαδή είναι ένας τρόπος χρωματισμού των κορυφών γραφήματος με αποτέλεσμα καμία γειτονική κορυφή να έχει το ίδιο χρώμα. Παρομοίως, ένας χρωματισμός άκρου αποδίδει ένα χρώμα σε κάθε άκρη, με αποτέλεσμα δύο γειτονικές άκρες να μοιράζονται το ίδιο χρώμα. Ο χρωματισμός μια περιοχής ενός επίπεδου γραφήματος εκχωρεί ένα χρώμα σε κάθε περιοχή έτσι ώστε καμία περιοχή που μοιράζεται ένα όριο να έχει το ίδιο χρώμα.

Η σύμβαση χρήσης χρωμάτων προέρχεται από το χρωματισμό των χωρών ενός χάρτη, όπου κάθε πρόσωπο είναι κυριολεκτικά χρωματισμένο. Αυτό γενικεύτηκε για το χρωματισμό των προσώπων ενός γραφήματος που είναι ενσωματωμένο στο επίπεδο. Με την επίπεδη δυαδικότητα έγινε χρωματισμός των κορυφών και σε αυτήν τη μορφή γενικεύεται σε όλα τα γραφήματα. Σε μαθηματικές και υπολογιστικές αναπαραστάσεις, είναι τυπικό να χρησιμοποιούνται οι πρώτοι θετικοί ή μη αρνητικοί ακέραιοι ως "χρώματα". Σε γενικές γραμμές, μπορεί κανείς να χρησιμοποιήσει οποιοδήποτε πεπερασμένο σετ ως "σύνολο χρωμάτων". Η φύση του προβλήματος χρωματισμού εξαρτάται από τον αριθμό των χρωμάτων αλλά όχι από το τι είναι.

Παρακάτω βλέπουμε ένα παράδειγμα χρωματισμού γραφημάτων.



### 3. Προσεγγίσεις Επίλυσης

#### 3.1. Δεδομένα Προβλήματος (Toronto Datasets)

Τα δεδομένα που χρησιμοποιούνται αποτελούν μια απλοποιημένη μορφή του προβλήματος η οποία έχει προταθεί το 1996 από τους Carter κ.ά. Τα δεδομένα αποτελούνται από ένα σύνολο 13 πραγματικών προβλημάτων χρονοπρογραμματισμού και συγκεκριμένα από 3 Καναδικά γυμνάσια, 5 Καναδικά πανεπιστήμια, 1 Αμερικανικό πανεπιστήμιο, 1 Βρετανικό πανεπιστήμιο και ένα πανεπιστήμιο στην Σαουδική Αραβία.

Τα αρχεία δεδομένων διαθέτουν για κάθε σπουδαστή μια γραμμή που περιέχει τους αριθμούς των μαθημάτων στα οποία είναι εγγεγραμμένος χωρισμένους μεταξύ τους με κενά. Η πρώτη γραμμή του αρχείου αντιστοιχεί στον πρώτο σπουδαστή, η δεύτερη γραμμή στο δεύτερο σπουδαστή κ.ο.κ. Για παράδειγμα το αρχείο car-f-92.stu περιέχει 18419 σειρές δεδομένων και ξεκινά με τις ακόλουθες σειρές:

- 0170
- 0156
- 0281
- 0006
- 0154 0156
- 0383

που σημαίνουν ότι ο φοιτητής 1 έχει εγγραφεί στο μάθημα 0170, ο φοιτητής 2 έχει εγγραφεί στο μάθημα 0156, ο φοιτητής 3 έχει εγγραφεί στο μάθημα 0281, ο φοιτητής 4 έχει εγγραφεί στο μάθημα 0006, ο φοιτητής 5 στα μαθήματα 0154 0156 κ.ο.κ.

Πρόβλημα	Αρχείο Δεδομένων	Εξετάσεις	Φοιτητές	Εγγραφές	Περίοδοι	Πυκνότητα
car-f-92	car-f-92.stu	543	18419	55522	32	0.14
car-s-91	car-s-91.stu	682	16925	56877	35	0.13
ear-f-83	ear-f-83.stu	190	1125	8109	24	0.27
hec-s-92	hec-s-92.stu	81	2823	10632	18	0.42
kfu-s-93	kfu-s-93.stu	461	5349	25113	20	0.06
lse-f-91	lse-f-91.stu	381	2726	10918	18	0.06
pur-s-93	pur-s-93.stu	2419	30029	120681	42	0.03
rye-s-93	rye-s-93.stu	486	11483	45051	23	0.07
sta-f-83	sta-f-83.stu	139	611	5751	13	0.14
tre-s-92	tre-s-92.stu	261	4360	14901	23	0.18
uta-s-92	uta-s-92.stu	622	21266	58979	35	0.13
ute-s-92	ute-s-92.stu	184	2749	11793	10	0.08
yor-f-83	yor-f-83.stu	181	941	6034	21	0.29

### 3.2. Περιγραφή του Προγράμματος

Εφόσον κατεβάσουμε το αρχείο της εφαρμογής και το τρέξουμε θα δούμε το menu επιλογών της εφαρμογής. Το menu από αποτελείτε από τις εξής επιλογές:

- **Load Data:** Έπειτα από τη φόρτωση των προβλημάτων εμφανίζει στο χρήστη πληροφορίες τους προβλήματος, για παράδειγμα πυκνότητα, εξετάσεις, αριθμό φοιτητών.
- **Check Solution:** Γίνεται επίλυση του προβλήματος που έχει φορτωθεί και εμφανίζει αριθμό περιόδων και κόστος λύσης. Επίσης δημιουργεί αρχείο λύσης όπου περιέχει τις εξετάσεις.
- **Solve Problem:** Γίνεται έλεγχος ορθότητας ως προς τον αριθμό συγκρούσεων και ως προς τον αριθμό εξετάσεων.
- **Solve All:** Εκτελεί όλες τις διαδικασίες και επιλύει όλα τα στιγμιότυπα των προβλημάτων.
- **Exit:** κλείνει το πρόγραμμα.

#### 3.2.1. Σύντομη Περιγραφή του Κώδικα

Ο κώδικας χωρίζεται σε συναρτήσεις κάθε μια από αυτές υλοποιεί μια συγκεκριμένη λειτουργία που μας ζητάτε από την εκφώνηση της εργασίας.

**LoadProblem:** Διαβάζει το αρχείο και δημιουργεί ένα πίνακα συγκρούσεων και υπολογίζει τη πυκνότητα του προβλήματος με τους αριθμούς των φοιτητών οι οποίοι είναι εγγεγραμμένοι σε αυτό.

**Graph:** Επιλύει το τρέχον πρόβλημα με την χρήση της μεθόδου χρωματισμού γράφων (graph coloring).

**SolveProblem:** Υπολογίζει το κόστος της λύσης του προβλήματος με βάση τις συγκρούσεις.

**SolveAll:** Δημιουργεί ένα νέο αρχείο όπου εκεί εκχωρεί τη λύση του προβλήματος.



## 4. Επίλυση Προβλήματος

### 4.1. First Fit

Στη συγκεκριμένη εργασίας χρησιμοποιήθηκε ο αλγόριθμος First Fit ο οποίος ξεκινά και σαρώνει το πίνακα από την αρχή και επιλέγει το πρώτο διαθέσιμο μπλοκ που είναι αρκετά μεγάλο. Είναι ένας ταχύτατος αλγόριθμος προσπέλασης. Τέλος, στο πρόγραμμα θα μας βοηθήσει στη φόρτωση πολλών δεδομένων στο εμπρός τμήμα της μνήμης που θα πρέπει να εξετάζουμε κάθε φορά που υπάρχουν δεδομένα στο πίνακα.

Παρακάτω θα δούμε πληροφορίες που χρησιμοποιήσαμε για αυτή την τεχνική.

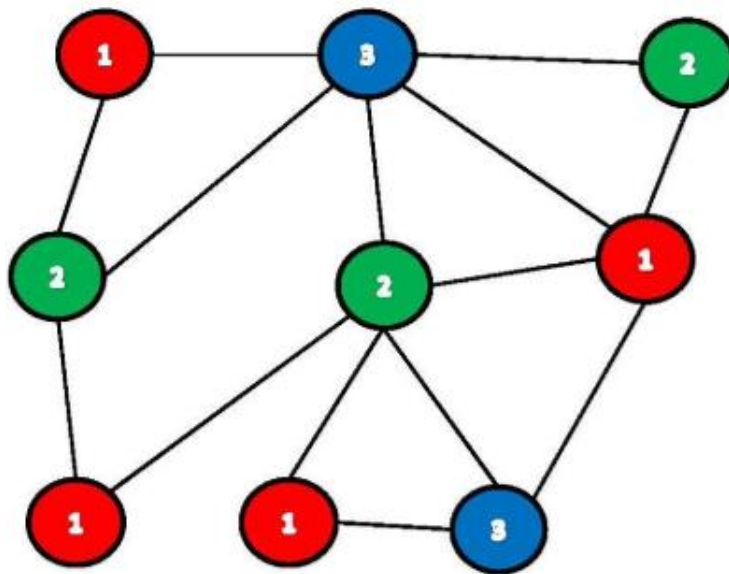
### Greedy Colouring (First Fit)

- $\pi$  = permutation (διάταξη κορυφών)
- Independent set (ανεξάρτητο σύνολο) = σύνολο κορυφών που δεν συνδέονται μεταξύ τους με ακμές
- $O(|V|^2)$

GREEDY ( $S \leftarrow \emptyset, \pi$ )	
(1)	<b>for</b> $i \leftarrow 1$ <b>to</b> $ \pi $ <b>do</b>
(2)	<b>for</b> $j \leftarrow 1$ <b>to</b> $ S $
(3)	<b>if</b> $(S_j \cup \{\pi_i\})$ is an independent set <b>then</b>
(4)	$S_j \leftarrow S_j \cup \{\pi_i\}$
(5)	<b>break</b>
(6)	<b>else</b> $j \leftarrow j + 1$
(7)	<b>if</b> $j >  S $ <b>then</b>
(8)	$S_j \leftarrow \{\pi_i\}$
(9)	$S \leftarrow S \cup S_j$

### 4.2. Greedy Graph Coloring

Για την επίλυση του προβλήματος χρησιμοποιήσαμε τον αλγόριθμο **Greedy Graph Coloring**. Πιο συγκεκριμένα δημιουργήσαμε πρόγραμμα εξετάσεων για ένα πανεπιστήμιο. Ως αποτέλεσμα θα έχουμε λίστα διαφορετικών θεμάτων και μαθητών που εγγράφονται σε κάθε μάθημα. Πολλά μαθήματα θα έχουν κοινούς μαθητές. Έτσι θα προγραμματίσουμε τις εξετάσεις ώστε να μην προγραμματίζονται ταυτόχρονα δύο εξετάσεις με έναν κοινό μαθητή και πόση πυκνότητα απαιτεί για το κάθε αρχείο δεδομένων. Αυτό το πρόβλημα μπορεί να αναπαρασταθεί ως γράφημα όπου κάθε κορυφή είναι ένα θέμα και μια ακμή μεταξύ δύο κορυφών σημαίνει ότι υπάρχει ένας κοινός μαθητής. Αυτό είναι λοιπόν ένα πρόβλημα χρωματισμού γραφήματος όπου ο ελάχιστος αριθμός χρονικών χρόνων ισούται με τον χρωματικό αριθμό του γραφήματος.



- **Chromatic Number:** Ο μικρότερος αριθμός χρωμάτων που απαιτούνται για να χρωματίσει ένα γράφημα  $G$  ονομάζεται χρωματικός αριθμός.
- **Vertex coloring:** είναι το σημείο εκκίνησης του θέματος και άλλα προβλήματα χρωματισμού μπορούν να μετατραπούν σε έκδοση κορυφής.

## 5. Αποτελέσματα

Ως αποτέλεσμα της παρούσας εργασίας υλοποιήθηκε με τη γλώσσα προγραμματισμού Java. Έχει δημιουργηθεί το απαραίτητο Graphic User Interface όπου από εκεί θα επιλέγει ο χρήστης την ανάλογη εργασίας που θα επιθυμεί. Να φορτώνει τα δεδομένα και να τα φέρνει ως αποτέλεσμα στην οθόνη. Να γίνεται προσέγγιση της λύσης των δεδομένων και να δίνει πιθανές λύσεις και επίλυση όλων αυτών των δεδομένων.

## 6. Συμπεράσματα

Συμπεράσματα δεν μπόρεσα να συλλέξω διότι δεν κατάφερα να τελειώσω την εργασία μέχρι το τελικό στάδιο, ως κύριο λόγω το φόρτο εργασίας και έλλειψη χρόνου, με αποτέλεσμα να μη δουλεύουν σωστά τα κομμάτια που έχω συνδέσει.

## Αναφορές

1. <https://iq.opengenus.org/graph-colouring-greedy-algorithm/>
2. [https://en.wikipedia.org/wiki/Graph\\_coloring](https://en.wikipedia.org/wiki/Graph_coloring)
3. <https://www.geeksforgeeks.org/m-coloring-problem-backtracking-5/>
4. <https://www.geeksforgeeks.org/graph-coloring-set-2-greedy-algorithm/>
5. <https://github.com/chgogos/alco>
6. [https://en.wikipedia.org/wiki/Graph\\_coloring](https://en.wikipedia.org/wiki/Graph_coloring)
7. <https://www.geeksforgeeks.org/program-first-fit-algorithm-memory-management/>