ΧΡΩΜΑΤΙΣΜΟΣ ΓΡΑΦΩΝ

### **ΤΜΗΜΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΟΙΚΙΝΩΝΙΩΝ ΠΑΝΕΠΙΣΤΗΜΙΟΥ ΙΩΑΝΝΙΝΩΝ**

ΟΝ/ΜΟ:Νάστος Βασίλειος

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ:Χρήστος Γκόγκος

**ΠΕΡΙΕΧΟΜΕΝΑ**

[**ΤΜΗΜΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΟΙΚΙΝΩΝΙΩΝ ΠΑΝΕΠΙΣΤΗΜΙΟΥ ΙΩΑΝΝΙΝΩΝ** 1](#_Toc59109291)

[ΠΕΡΙΛΗΨΗ 1](#_Toc59109292)

[1.Εισαγωγή στα Np Προβλήματα 2](#_Toc59109293)

[1.1.P vs NP 2](#_Toc59109294)

[1.2.NP COMPLETE Προβλήματα 2](#_Toc59109295)

[2.Περιγραφή του προβλήματος χρωματισμού γράφων. 3](#_Toc59109296)

[3.Προσεγγίσεις Επίλυσης 3](#_Toc59109297)

[3.1.Δεδομένα Προβλήματος(Toronto DataSet) 3](#_Toc59109298)

[3.2 ΠΙΝΑΚΑΣ ΣΤΑΤΙΣΤΙΚΩΝ ΣΤΟΙΧΕΙΩΝ ΠΡΟΒΛΗΜΑΤΩΝ 4](#_Toc59109299)

[4.Αλγόριθμος First\_Fit 4](#_Toc59109300)

[5.Αλγόριθμος DSatur 5](#_Toc59109301)

[6.Αλγόριθμος RLF(Recursive Largest First) 6](#_Toc59109302)

[7.Αλγόριθμος BACKTRACKING DSATUR 7](#_Toc59109303)

[ΣΥΜΠΕΡΑΣΜΑΤΑ 7](#_Toc59109304)

[ΑΝΑΦΟΡΕΣ 8](#_Toc59109305)

# ΠΕΡΙΛΗΨΗ

1.Εισαγωγή στα Np Προβλήματα.

Τα Np προβλήματα αφορούν προβλήματα υλοποιήσης σε μη πολυωνυμικό χρόνο.Tα προβλήματα Np-Complete είναι στην ουσία,τα δυσκολότερα προβλήματα της κλάσης NP,τα οποία αφορούν προβλήματα χωρίς γνωστό αποδοτικό πολυωνυμικό αλγόριθμο.Ένα πρόβλημα Np-Complete είναι ένα πρόβλημα στο οποίο μετασχηματίζεται πολυωνυμικά κάθε άλλο πρόβλημα της κλάσης NP.Αν γνωρίζουμε τον αλγόριθμο για ένα πρόβλημα Np-Complete,μπορούμε να επιλύσουμε κάθε άλλο πρόβλημα της NP.Γνωστό NP\_Complete πρόβλημα είναι το πρόβλημα ελέγχου ικανοποιησιμότητας λογικών εκφράσεων.Τα ΝP-Hard προβλήματα αφορούν προβλήματα που η λύση τους δεν υλοποιήται σε πολυωνυμικό χρόνο,ο βαθμός δυσκολίας είναι ίδιος με τα NP-Προβλήματα,ωστόσο τα Np-Hard προβλήματα δεν πρέπει να είναι NP.Γνωστό ΝP-Hard πρόβλημα είναι το πρόβλημα του πλανώδιου πωλητή.Γενικά κάθε πρόβλημα NP μπορεί να επιβεβαιωθεί σε πολυωνυμικό χρόνο από έναν αλγόριθμο που ενδιάμεσα από τις διαθέσιμες επιλογές θα επιλέγει πάντα την σωστή(«Lucky algorithm»).

## 1.1.P vs NP

Το πρόβλημα στην απλή του διατύπωση θέτει το ερώτημα αν μπορεί ένα πρόβλημα να επιλυθεί τόσο γρήγορα από τον υπολογιστή όσο γρήγορα μπορεί να επιβεβαιωθεί η ύπαρξη της λύσης του.Το πρόβλημα P vs NP είναι ένα ανοικτό πρόβλημα στην επις΄τημη των υπολογιστών.P είναι η γενική κλάση των ερωτημάτων των προβλημάτων που υπάρχει αλγόριθμος ο οποίος επιλύει το πρόβλημα σε πολυωνιμικό χρόνο.Αντίστοιχα με τον όρο NP αναφέρομαστε στην κλάση των προβλημάτων για τα οποία η λύση δεν μπορεί να βρεθεί σε πολυωνυμικό χρόν,ωστό σο το πρόβλημα μπορεί να επιβεβαιωθεί σε πολυωνιμικό χρόνο,δηλαδή η απάντηση μπορεί να επιβεβαιωθεί γρήγορα. (tutorialspoint, 2020)

## 

## 1.2.NP COMPLETE Προβλήματα

Ας υποθέσουμε ότι έχουμε 2 προβλήματα απόφασης L1 και L2,και έναν αλγόριθμο Α2 που λύνει το πρόβλημα L1.Για να μπορέσουμε να επιτύχουμε μείωση(Reduction) πρέπει να μετασχηματίσουμε τα προβλήματα L1 και L2 ώστε ο αλγόριθμος A2 να αποτελεί μέρος του αλγόριθμου A1 που θα χρησιμοποιήται για την επίλυση του προβλήματος L1.Για να μπορέσω να αναφερθώ σε ένα πρόβλημα(L) ως NP-COMPLETE πρέπει :

* Το πρόβλημα να ανήκει στην κλάση Νp.
* Κάθε πρόβλημα Np πρέπει να μπορεί να αναλώνεται στο πρόβλημα L(Reduction).

Η προυπόθεση για να θεωρηθεί ένα πρόβλημα(L) σαν NP-Complete είναι,ότι κάθε πρόβλημα ΝP να μπορεί να χρησιμοποιήσει το για την επιβεβαίωση του σε πολυωνιμικό χρόνο.To πρώτο ΝP-COMPLETE πρόβλημα που διατυπώθηκε ήταν το SAT .Η γνώση γύρω από τα Np προβλήματα και συγκεκριμένα τα Np-Complete είναι σημαντική γιατί για ένα πρόβλημα μπορούμε να αποδείξουμε ότι η λύση σε πολυωνυμικό χρόνο η λύση δεν μπορεί να είναι εφικτή.[[1]](#footnote-1)

# 2.Περιγραφή του προβλήματος χρωματισμού γράφων.

Το πρόβλημα χρωματισμού γραφήματος τυπικά ορίζεται ως εξής. Δεδομένου ενός μη κατευθυνόμενου απλού γραφήματος G = (V, E) με ένα σύνολο κορυφών V και ένα σύνολο ακμών E, ζητείται η ανάθεση σε κάθε κορυφή v ∈ V ενός ακεραίου c(v) ∈ {1, 2, ..., k} έτσι ώστε το k να ελαχιστοποιείται και να ισχύει ότι c(v) ≠ c(u) ∀{v, u} ∈ E.Το πρόβλημα συναντάται σε μεγάλο αριθμό πρακτικών εφαρμογών όπως ο χρονοπρογραμματισμός εκπαιδευτικών ιδρυμάτων (educational timetabling), ο χρονοπογραμματισμός αθλητικών γεγονότων (sports scheduling), η ανάθεση συχνοτήτων (frequency assignment), η ανάθεση καταχωρητών στους μεταγλωττιστές (compiler register allocation) και άλλα.Πολλοί αλγόριθμοι χρωματισμού γραφημάτων έχουν προταθεί τα τελευταία 50 έτη. Στην παρούσα εργασία θα εξεταστούν τέσσερις αλγόριθμοι που ανήκουν στις λεγόμενες κατασκευαστικές τεχνικές (constructive techniques). Οι κατασκευαστικές τεχνικές δημιουργούν λύσεις βήμα προς βήμα, αναθέτοντας στη σειρά, σε κάθε κορυφή, ένα χρώμα, πιθανά εφαρμόζοντας οπισθοχώρηση κατά τη διαδικασία. Οι αλγόριθμοι που θα εξεταστούν είναι ο αλγόριθμος first fit, ο αλγόριθμος DSATUR, ο αλγόριθμος Recursive Largest First και ο αλγόριθμος backtracking DSATUR.Τα δεδομένα θα χωριστούν σε χρωματικές τάξεις.Μια χρωματική τάξη αποτελείται από κορυφές που έχουν χρωματιστεί με το ίδιο χρώμα.Οι αλγόριθμοι χρωματισμού προσπαθούν να επιτύχουν χρωματισμό γράγων με όσο το δυνατόν λιγότερο αριθμό χρωμάτων.

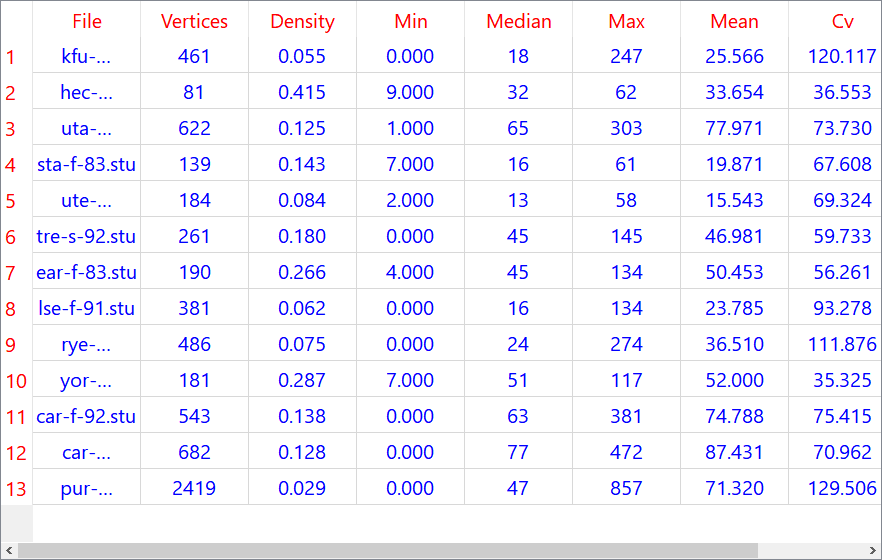
# 3.Προσεγγίσεις Επίλυσης

## 3.1.Δεδομένα Προβλήματος(Toronto DataSet)

Τα δεδομένα του προβλήματος βρίσκονται στον σύνδεσμο:dataset.Το αρχείο περιέχει 13 αρχεία δεδομένων,όπου σε κάθε γραμμή αναπαρίσταται και ένας φοιτητής ενώ σε κάθε στήλη διαχωρισμένoi με το κενό βρίσκονται οι κωδικοί εξέτασης που έχει εγγραφεί ο κάθε φοιτητής.Κάθε αρχείο παρέχει και ένα μοναδικό αριθμό απο κορυφές,δηλαδή το σύνολο των μοναδικών διανυσμάτων που συμμετέχουν οι εγγεγραμένοι φοιτητές.Σε κάθε αρχείο υπάρχει και ένας μοναδικός αριθμός διαγωνισμάτων.Σκοπός είναι ο χρωματισμός των κορυφών(δηλαδή η ομαδοποίηση των διαγωνισμάτων),με όσο το δυνατόν λιγότερα χρώματα.

## 3.2 ΠΙΝΑΚΑΣ ΣΤΑΤΙΣΤΙΚΩΝ ΣΤΟΙΧΕΙΩΝ ΠΡΟΒΛΗΜΑΤΩΝ

Στην παρακάτω εικόνα εμφανίζονται τα σταταστικά στοιχεία που προκύπτουν σε κάθε πρόβλημα.Τα ζητούμενα χαρακτηριστικά είναι ο αριθμός των κορυφών(V),η πυκνότητα των συγκρούσεων(Density),η διάμεσος κορυφή(Med),η κορυφή με τον μικρότερο βαθμό (Min),η κορυφή με τον μεγαλύτερο βαθμό(Max),o μέσος βαθμός μεταξύ των κορυφων(Mean),και ο συντελεστής διακύμανσης(Cv).Τα παρακάτω δεδομένα συγκεντρώθηκαν και από τα 13 αρχεία του προβλήματος.



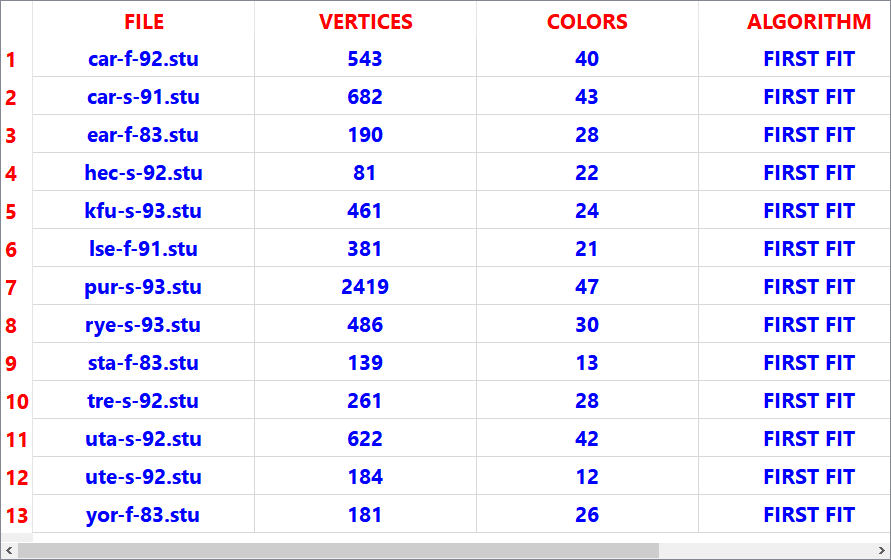
Εικόνα.Στατιστικά δεδομένα αρχείων

# 4.Αλγόριθμος First\_Fit

Ο αλγόριθμος first -fit είναι ένας greedy(άπληστος) αλγόριθμος χρωματισμού κορυφών.Για κάθε κορυφή του γραφήματος μας ο αλγόριθμος ακολουθεί τα εξής βήματα:

* Καταγραφή χρωμάτων γειτονικών κορυφών,εφόσον έχουν περάσει την διαδικασία χρωματισμού,σαν μη διαθέσιμα.
* Εύρεση του πρώτου διαθέσιμου χρώματος.
* Χρωματισμός κορυφής με το διαθέσιμο χρώμα.
* Επαναφορά διαθεσιμότητας χρωμάτων σε κατάσταση διαθέσιμα,για να χρησιμοποίηση τους στις επόμενες κορυφές.

Ο αλγόριθμος first-fit έχει σαν κύριο χαρακτηριστικό ότι χρωματίζει τις κορυφές με την σειρά ξεκινώντας από την πρώτη και χωρίς να ενσωματώνει σε κάποιο στάδιο τουχαρακτηριστικά των κορυφών όπως ο βαθμός μίας κορυφής ή το βάρος της κορυφής.Το χαρακτηριστικό αυτό των καθιστά σαν έναν άπληστο αλγόριθμο.Στην παρακάτω εικόνα φαίνονται τα αποτελέσματα που θα παραχθούν με βάση τα δεδομένα του προβλήματος μας από το αρχείο δεδομένων Toronto αν για κάθε γράφημα που παράγουν τα αρχεία εκτελεστεί και παράξει αποτελέσματα ο αλγόριθμος first-fit.Ο αλγόριθμος first fit έχει πολυπλοκότητα O(V+E).Αυτό σημαίνει ότι ο χρόνος εκτέλεσης του εξαρτάται από το μέγεθος του γραφήματος μας και Από τον αριθμό διασυνδέσεων που προκύπτουν μεταξύ των κόμβων.

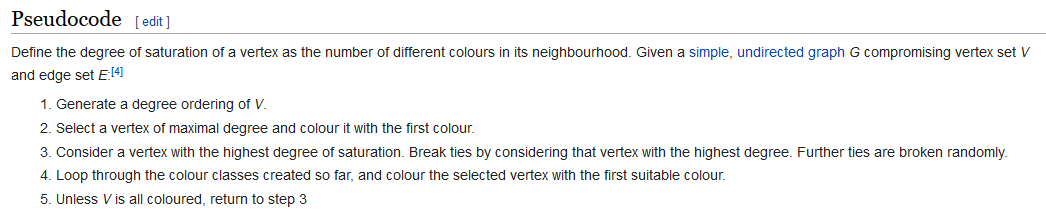


ΕΙΚΟΝΑ 3.ΑΠΟΤΕΛΕΣΜΑΤΑ ΑΛΓΟΡΙΘΜΟΥ FIRST\_FIT([[2]](#footnote-2))

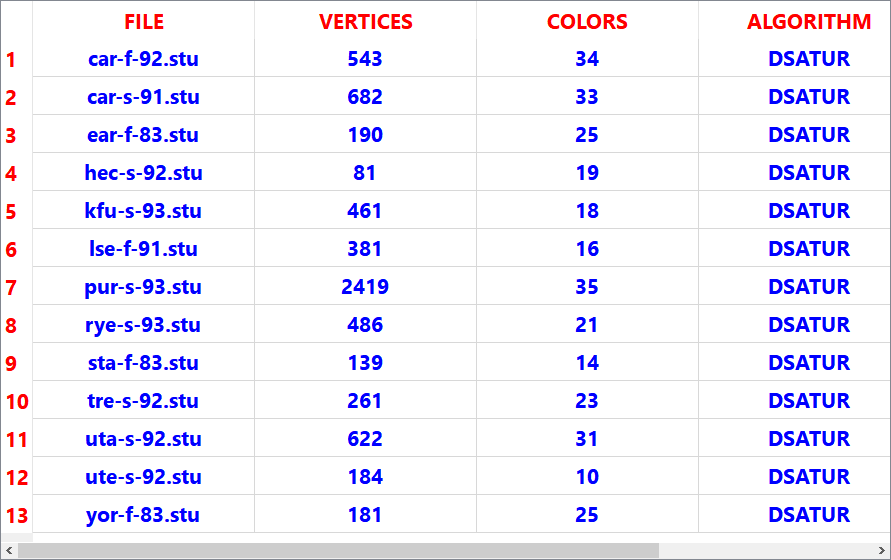
# 5.Αλγόριθμος DSatur

Ο αλγόριθμος DSATUR,είναι ένας ευρετικός αλγόριθμος χρωματισμού γράφων.Ο αλγόριθμος επιλέγει την κορυφή με τον μεγαλύτερο βαθμό και την χρωματίζει με το πρώτο χρώμα.Στην συνέχεια επιλέγεται η κορυφή με τον μεγαλύτερο βαθμό κορεσμού.Αν δύο κορυφές έχουν τον ίδιο βαθμό κορεσμού επιλέγεται η κορυφή με το μεγαλύτερο βαθμό.Παραπάνω ισότητες διαχωρίζονται με τυχαίο τρόπο.Έπειτα με χρήση επανάληψης έλεγχουμε από τις γειτονικές κορυφές της επιλεγμένης κορυφής ,το χρώμα που έχουν χρωματιστεί και βρίσκουμε το κατάλληλο χρώμα για την κορυφή.Αν όλες οι κορυφές έχουν χρωματιστεί ο αλγόριθμος τερματίζει αλλιώς βρίσκει την επόμενη κορυφή με τον μεγαλύτερο βαθμό κορεσμού και εκτελεί τα επόμενα βήματα ,εώς ότου ολοκληρωθεί ο χρωματισμός όλων των κορυφών του γραφήματος.Η πολυπλοκότητα του αλγορίθμου DSatur στην χειρότερη περίπτωση είναι O(n^2).Παρατηρώντας το αποτελέσματα που παράγει ο DSatur συγκριτικά με τον First Fit,προκύπτει το συμπέρασμα ότι ο αλγόριθμος παράγει καλύτερα αποτελέσματα χρωματισμού από ότι ο first fit και γενικά οι greedy coloring αλγόριθμοι.

Πηγή (Hemert, 2006)



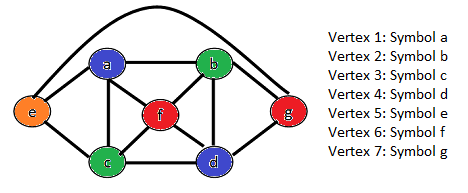
Εικόνα 4.Ψευδοκώδικας για υλοποιήση DSATUR



ΕΙΚΟΝΑ 5.ΑΠΟΤΕΛΕΣΜΑΤΑ ΑΛΓΟΡΙΘΜΟΥ DSATUR([[3]](#endnote-1))

Ο αλγόριθμος DSATUR

# 6.Αλγόριθμος RLF(Recursive Largest First)



Εικόνα 1.Παράδειγμα εκτέλεσης Αλγορίθμου RLF (rlf icon image, n.d.)

Ο αλγόριθμος RLF, σε κάθε βήμα της διαδικασίας επιλέγεi ένα κόμβο για χρωματισμό που, χρωματίζοντας όλες τις κορυφές με τα λιγότερα δυνατά χρώματα.Ο αλγόριθμος RLF υλοποιήται ως εξής:Δοθέντως ενός γράφου G με κορυφές V και ακμές E,ο αλγόριθμος αναθέτει το χρώμα 1(η 0),στην κορυφή με τον μέγιστο βαθμό,υποθετικά η u1.Το χρώμα που χρωματίζει αρχικά ο αλγόριθμος είναι i.Όταν i κόμβοι χρωματιστούν με το χρώμα ο αλγόριθμος τοποθετεί σε ένα σύνολο U1 τους κόμβους που είναι γείτονες με έναν τουλάχιστον χρωματισμένο κόμβο και δεν έχουν χρωματιστεί,και σε ένα σύνολο U2 τους κόμβους που δεν είναι γείτονες με κανέναν από τους χρωματισμένους κόμβους και επιλέγεται η κορυφή με τον ελάχιστο βαθμό από το σύνολο U1,αν δεν υπάρχεί.Αν δεν υπάρχει κάποια διαθέσιμη επιλογή(σ.σ το σύνολο είναι άδειο),ο αλγόριθμος εκτελείται ανδρομικά για το υπόλοιπο τμήμα του γράφου,συμπεριλαμβανομένων και των μη χρωματισμένων κορυφών στις οποίες δεν υπήρχε γειτονικότητα με χρωματισμένες κορυφές,χρησιμοποιώντας το επόμενο διαθέσιμο χρώμα. (Leighton, 1979)

Στην παρακάτω εικόνα ακολουθεί ένα παράδειγμα εκτέλεσης του αλγορίθμου Rlf,για το αρχείο δεδομένων sta-f-83.stu.Ο αλγόριθμος επιτυνχάνει να χρωματίσει 139 κορυφές χρησιμοποιώντας δεκατρία χρώματα.Στην παρακάτω εικόνα φαίνονται τα αποτελέσματα εκτέλεσης του αλγορίθμου RLF.

# 7.Αλγόριθμος BACKTRACKING DSATUR

Ψευδοκώδικας για backtracking.

**procedure** bt(c) **is**

**if** reject(P, c) **then** return

**if** accept(P, c) **then** output(P, c)

s ← first(P, c)

**while** s ≠ NULL **do**

bt(s)

s ← next(P, s)

# ΣΥΜΠΕΡΑΣΜΑΤΑ

# ΑΝΑΦΟΡΕΣ

Hemert, I. ́. J.-J. I. v., 2006. *jsoftware.* [Ηλεκτρονικό]   
Available at: http://www.jsoftware.us/vol1/jsw0102-03.pdf  
[Πρόσβαση 8 December 2020].

Leighton, F. T., 1979. [Ηλεκτρονικό]   
Available at: https://pdfs.semanticscholar.org/128d/490e1f116b410e4fd2482b54c742eb8d4371.pdf  
[Πρόσβαση 29 Νοέμβριος 2020].

rlf icon image, χ.χ. [Ηλεκτρονικό]   
Available at: https://www.codeproject.com/KB/recipes/graph\_coloring\_using\_RLF/gcq\_3.png  
[Πρόσβαση 29 Νοέμβριος 2020].

tutorialspoint, 2020. *tutorialspoint.* [Ηλεκτρονικό]   
Available at: https://www.tutorialspoint.com/design\_and\_analysis\_of\_algorithms/design\_and\_analysis\_of\_algorithms\_np\_hard\_complete\_classes.htm  
[Πρόσβαση Tuesday 12 2020].

1. https://www.geeksforgeeks.org/np-completeness-set-1/ [↑](#footnote-ref-1)
2. <https://github.com/vasnastos/Algorithms_and_complexity/blob/main/Alco_report_images/FIRST_FIT.png>

   [↑](#footnote-ref-2)
3. <https://raw.githubusercontent.com/vasnastos/Algorithms_and_complexity/main/Alco_report_images/DSATUR.png?token=APD2HAMOXBUKJN7742ONFG274SFZW>

   [↑](#endnote-ref-1)