ΕΠΛ236 Report

Υπολογιστικό σύστημα:

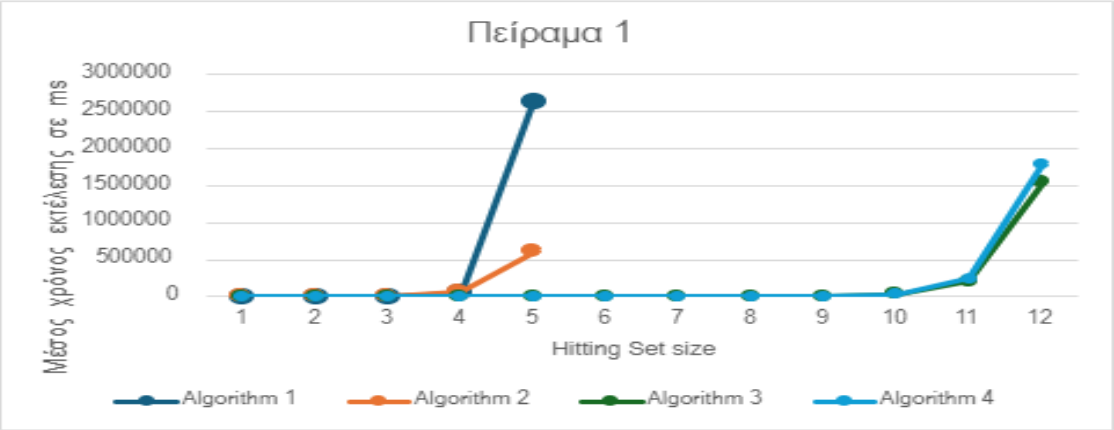
- Λειτουργικό Σύστημα: Microsoft Windows 11 Pro
- Τύπος Επεξεργαστή: 12th Gen Intel(R) Core(TM) i7-1255U
- Ταχύτητα Επεξεργαστή: 1700 MHz
- Πλήθος Πυρήνων: 10
- Μνήμη RAM: 16 GB

Πείραμα 1

Σενάριο πειράματος:

Ένα ιδανικό στιγμιότυπο εισόδου για να μην υπάρχει σύνολο κρούσης για μικρά k, είναι εκείνο για το οποίο ισχύουν οι παρακάτω προϋποθέσεις:

1) Υπάρχουν αρκετά στοιχεία τα οποία είναι μοναδικά σε ορισμένα υποσύνολα, έτσι ώστε να είναι σίγουρη η αποτυχία εύρεσης συνόλου κρούσης για μικρότερες τιμές του k.
2) Μεγάλος αριθμός υποσυνόλων (m) για την αύξηση της πιθανότητας να παράγονται περισσότερα μοναδικά στοιχεία.
3) Μικρός αριθμός μέγιστου μεγέθους υποσυνόλων (c) σε σύγκριση με το πλήθος όλων των πιθανών τιμών στοιχείων (n) για την μείωση πιθανότητας παραγωγής κοινών στοιχείων σε κάθε υποσύνολο.
   Πειραματικά Αποτελέσματα: Παρακάτω δίδεται το γράφημα για τιμές n = 500, m = 100, c = 100:



Η σειρά αποδοτικότητας των αλγορίθμων από τον χειρότερο προς τον καλύτερο είναι: 1ος, 2ος, 4ος και 3ος. Αρχικά, είναι αξιοσημείωτο να αναφέρω πως στην ασυμπτωτική μου ανάλυση χρονικής πολυπλοκότητας ξέχασα να λάβω υπόψη την αναδρομική φύση των αλγορίθμων, το οποίο σημαίνει πως η πολυπλοκότητα τους είναι O(kc^(k+1)m) το οποίο εξηγά την ξαφνική αύξηση στον μέσο χρόνο που απαιτείται για τον υπολογισμό του συνόλου κρούσης από k = 4 στο k = 5 για τους αλγόριθμους 1 και 2, και από k = 11 στο k = 12 για τους αλγόριθμους 3 και 4.

Η πρόβλεψη μου για τον 1ο αλγόριθμο, ότι θα είναι ο λιγότερο αποτελεσματικός και κατ' επέκταση θα τερματίζει ως το k με την μικρότερη τιμή (εδώ 5), ήταν σωστή. Τα αποτελέσματα είναι αναμενόμενα διότι παρόλο της εξοικονόμησης χρόνου επιλογής, είναι

αρκετά πιθανό για τον αλγόριθμο αυτό να χρειάζεται πολύ περισσότερος χρόνος για να αποφασίσει ότι δεν υπάρχει σύνολο κρούσης λόγω των πάρα πολλών υποσυνόλων και στοιχείων. Αυτό το γεγονός οδηγεί σε μεγάλες πιθανότητες η τυχαία επιλογή όχι μόνο υποσυνόλου, αλλά και στοιχείου να ακολουθήσει σε χρονοβόρες επιλογές.

Ο 2ος αλγόριθμος επίσης τερματίζει ως το k = 5, κάτι που δεν περίμενα στην πρόβλεψή μου. Αυτά τα αποτελέσματα μετά από την πειραματική διαδικασία, μου φαίνονται λογικά, αφού παρόλο του πλεονεκτήματος επιλογής κρισιμότερου στοιχείου, η τεράστια ποσότητα και μοναδικότητα των υποσυνόλων που παράγονται δεν επιτρέπουν την διαγραφή πολλών υποσυνόλων, κάτι στο οποίο βασίζεται ο 2ος αλγόριθμος για την αποδοτικότητά του. Αυτό δεν του επιτρέπει να υπολογίσει μεγαλύτερη τιμή του k προτού περάσει μία ώρα εκτέλεσης. Ωστόσο, αν και η διαδικασία εύρεσης κρισιμότητας κάθε στοιχείου είναι χρονοβόρα (O(c^2 m)), εξοικονομά περισσότερο χρόνο με την επιλογή κρισιμότερων στοιχείων κατεβάζοντας αρκετά τον μέσο χρόνο εκτέλεσης σε σύγκριση με τον εντελώς τυχαιοποιημένο τρόπο επιλογής του 1ου αλγόριθμου.

Για τον 3ο αλγόριθμο, δεν περίμενα να είναι ο πιο αποτελεσματικός εφόσον ο 4ος θεωρητικά έχει τα πλεονεκτήματα και του 2ου και του 3ου. Παρόλο που τερματίζουν μέχρι το ίδιο k (εδώ 12) ο 3ος έχει ελαχίστως μικρότερο μέσο χρόνο εκτέλεσης. Αυτό το αποτέλεσμα με περαιτέρω ανάλυση, είναι λογικό. Στη χείριστη περίπτωση, όλα τα στοιχεία του επιλεγόμενου υποσυνόλου πρέπει να ελέγχονται σε κάθε αναδρομή. Αυτή η δυσκολία μειώνεται με την στοχευμένη επιλογή του αλγόριθμου 3, να επιλέγεται το μικρότερο σε μέγεθος υποσύνολο σε κάθε αναδρομή. Έτσι ελαχιστοποιείται ο χρόνος που περνά ανά αναδρομή. Φυσικά εάν ισχύει η προϋπόθεση που έθεσα στην πρόβλεψη μου, ο αλγόριθμος 3 γίνεται ακόμη πιο γρήγορος εφόσον εάν στα υποσύνολα που έχουν μικρό μήκος τυχαίνει να βρίσκονται τα πιο συχνά επαναλαμβανόμενα στοιχεία, θα ελαχιστοποιούνται οι συνολικές αναδρομές. Ο λόγος που ο 3ος αλγόριθμος είναι γρηγορότερος από τον 4ο είναι διότι ο απαιτούμενος χρόνος για την εύρεση κρισιμότητας κάθε στοιχείου είναι πολύ περισσότερος (O(c^2 m)) και ασύμφορος από την τυχαία επιλογή στοιχείου, ειδικά σε στιγμιότυπα με μεγάλο πλήθος υποσυνόλων και μέγιστου μεγέθους τους. Οι πιθανά χρονοβόρες επιλογές σε αυτή τη περίπτωση λόγω τυχαίας επιλογής είναι πολύ πιο περιορισμένες λόγω του μικρού μεγέθους του υποσυνόλου που επιλέγεται σε κάθε αναδρομή. Εφόσον η εύρεση κρισιμότητας πραγματοποιείται στον αλγόριθμο 4 και παραλείπεται στον αλγόριθμο 3, ο αλγόριθμος 3 τελικά γλυτώνει επιπρόσθετο χρόνο.

Συμπερασματικά, για τις τιμές των n, m και c που έχω δοκιμάσει αυτό το πείραμα και με βάση τις μετρήσεις που έλαβα, η σειρά αποδοτικότητας των αλγορίθμων όσο αφορά την ταχύτητα απόφασης ότι δεν υπάρχει σύνολο κρούσης για μικρές τιμές του k σε φθίνουσα σειρά είναι: 3ος, 4ος, 2ος και 1ος.

Πείραμα 2

Σενάριο πειράματος:

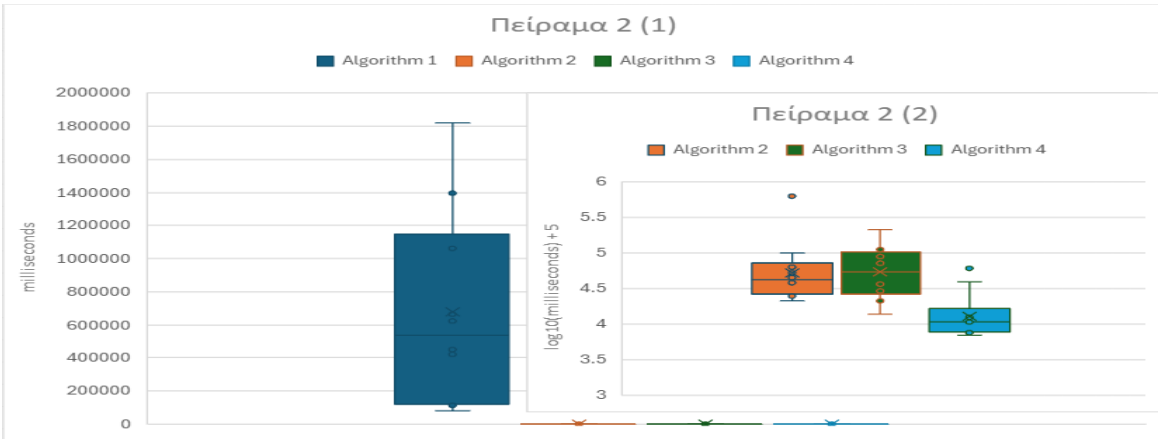
Ένα ιδανικό στιγμιότυπο εισόδου για να υπάρχει σύνολο κρούσης για συγκεκριμένο k αλλά ο εντοπισμός του να μην είναι άμεσος σε όλους τους αλγορίθμους είναι εκείνο για το οποίο ισχύουν οι παρακάτω προϋποθέσεις:

1) Τα τυχαία υποσύνολα την ώρα που παράγονται πρέπει να περιέχουν <= k μοναδικά στοιχεία τα οποία να καλύπτουν όλα τα υποσύνολα. Για παράδειγμα, για m = 100 και k = 5, το στιγμιότυπο πρέπει να περιέχει τα στοιχεία x1, x2, x3, x4 και x5 τα οποία διαμοιράζονται ομοιόμορφα στα 100 υποσύνολα με τρόπο που το κάθε υποσύνολο να περιέχει τουλάχιστον 1 από αυτά τα στοιχεία.

2) Μεγάλο μέγιστο πλήθος στοιχείων υποσυνόλων (c) για περαιτέρω καθυστέρηση στην εύρεση των στοιχείων που αναφέρονται στην 1η προϋπόθεση
3) Μεγάλο πλήθος δυνατών στοιχείων (n) για την μείωση πιθανότητας να παραχθούν πολλά στοιχεία που σιγουρεύουν την ύπαρξη συνόλου κρούσης για δεδομένη τιμή k. Η παραγωγή πολλών τέτοιων στοιχείων επιταχύνει υπερβολικά την διαδικασία εντοπισμού συνόλου κρούσης, κάτι που δεν επιθυμούμε στο συγκεκριμένο πείραμα. Ο τρόπος με τον οποίο παράγω τα στιγμιότυπα ήδη εξασφαλίζει την παραγωγή των στοιχείων που αναφέρονται στην 1η προϋπόθεση και άρα η παραγωγή περισσότερων είναι ανεπιθύμητη.

Πειραματικά Αποτελέσματα:

Παρακάτω δίδονται τα γραφήματα για τιμές n = 500, m = 100, c = 100, k = 5:



Παρατηρώ εδώ πως ο αλγόριθμος 1 έχει πολύ μεγαλύτερα άκρα (min και max) από τους άλλους αλγόριθμους στο σημείο που τα Box and Whisker plots τους εμφανίζονται σαν γραμμές στο γράφημα 1. Έτσι δημιούργησα το γράφημα 2 για τους αλγόριθμους 2, 3 και 4 όπου ο χρόνος εκτέλεσης εμφανίζεται σε log10(milliseconds) + 5 για λόγους εμφάνισης και για να μπορώ να σχολιάσω και να συγκρίνω τις μετρήσεις μου.

Τα αποτελέσματα του 1ου αλγόριθμου είναι λογικά αφού σαν τον λιγότερο αποτελεσματικό αλγόριθμο για λόγους που εξηγήθηκαν στο 1ο πείραμα, τα min, max, median, και γενικά όλα τα δεδομένα, έχουν πολύ μεγαλύτερες τιμές από τους υπόλοιπους αλγόριθμους. Η τεράστια διαφορά στις τιμές του 1ου με τους υπόλοιπους μπορεί να εξηγηθεί λόγο του γεγονότος ότι οποιαδήποτε έξυπνη επιλογή υποσυνόλου ή/και στοιχείου, εξοικονομά πολύ χρόνο εκτέλεσης, λόγω της τεράστιας ποσότητας υποσυνόλων που διαγράφονται λόγω της ύπαρξης στοιχείων που αναφέρονται στην προϋπόθεση 1.

Για τις μέγιστες τιμές, πρόσεξα πως οι αλγόριθμοι 2 και 4 έχουν χαμηλότερες από τον 3, παρόλο που ο 3 στο πείραμα 1 βγήκε ως ο πιο αποτελεσματικός. Αυτό σημαίνει πως η εύρεση κρισιμότητας στοιχείων εξοικονομεί περισσότερο χρόνο απ' ότι σπαταλά σε αυτή τη περίπτωση και για τις συγκεκριμένες τιμές των n, m, c και k. Το πολύ k στοιχεία (εδώ 5) στο κάθε υποσύνολο θα είναι η σωστή επιλογή για την εύρεση συνόλου κρούσης (άρα %5), και άρα η τυχαία επιλογή στοιχείου είναι πολύ πιθανότερο να είναι χρονοβόρα. Γι' αυτό το λόγο οι αλγόριθμοι 2 και 4 υπερτερούν στην μείωση μέγιστου χρόνου εκτέλεσης. Μεταξύ των 2 και 4, ο αλγόριθμος 4 περιορίζει περισσότερο την μέγιστη τιμή χρόνου, που είναι λογικό αφού η επιλογή μικρότερου υποσυνόλου σημαίνει πως θα βρεθεί ακόμα πιο γρήγορα το επιθυμητό στοιχείο.

Όσο αφορά τις διασπορές, ο αλγόριθμος 1 έχει την μεγαλύτερη, λόγο της αστάθειας

και τυχαιότητας των επιλογών του οι οποίες παράγουν πολύ απρόβλεπτες τιμές στο γράφημα. Ο αλγόριθμος με την 2η μεγαλύτερη διασπορά είναι ο 3ος. Αυτό οφείλεται στην τυχαία επιλογή στοιχείων η οποία επιλογή είναι πολύ πιθανό να οδηγεί σε αποτυχία όπως αναφέρεται πιο πάνω και άρα υπάρχει μια μικρή αστάθεια στις τιμές του στο γράφημα. Οι διασπορές των αλγόριθμων 2 και 4 είναι μικρές, που σημαίνει είναι πιο σταθεροί και προβλέψιμοι. Είναι περίπου οι ίδιες εφόσον λόγο της επιλογής κρισιμότερου στοιχείου πάντα θα επιλέγεται ένα από τα k επιθυμητά στοιχεία.

Τέλος, οι διάμεσοι του κάθε αλγόριθμου δείχνουν προς πια τιμή τείνουν να πλησιάζουν οι μετρήσεις του χρόνου εκτέλεσης. Η σειρά από μικρότερους σε μεγαλύτερους διάμεσους είναι: 4ος, 2ος, 3ος και 1ος η οποία σειρά είναι αναμενόμενη αφού σχετίζεται με την αποδοτικότητα του κάθε αλγόριθμου. Φαίνεται πως η σειρά των μέγιστων τιμών, των διασπορών και των διαμέσων είναι η ίδια.

Αυτό με φέρνει στο συμπέρασμα πως για τις τιμές των n, m, c και k που έχω δοκιμάσει αυτό το πείραμα, η σειρά αποδοτικότητας των αλγορίθμων για την εύρεση υπαρκτού συνόλου κρούσης σε φθίνουσα σειρά είναι: 4ος, 2ος, 3ος και 1ος.