ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ 1 - ΧΕΙΜΕΡΙΝΟ ΕΞΑΜΗΝΟ 2020-2021
ΕΡΓΑΣΙΑ 3
ΟΝΟΜΑΤΕΠΩΝΥΜΟ: ΣΚΕΥΟΦΥΛΑΚΑ ΜΑΡΙΑ
ΑΡΙΘΜΟΣ ΜΗΤΡΩΟΥ: 1115201900173

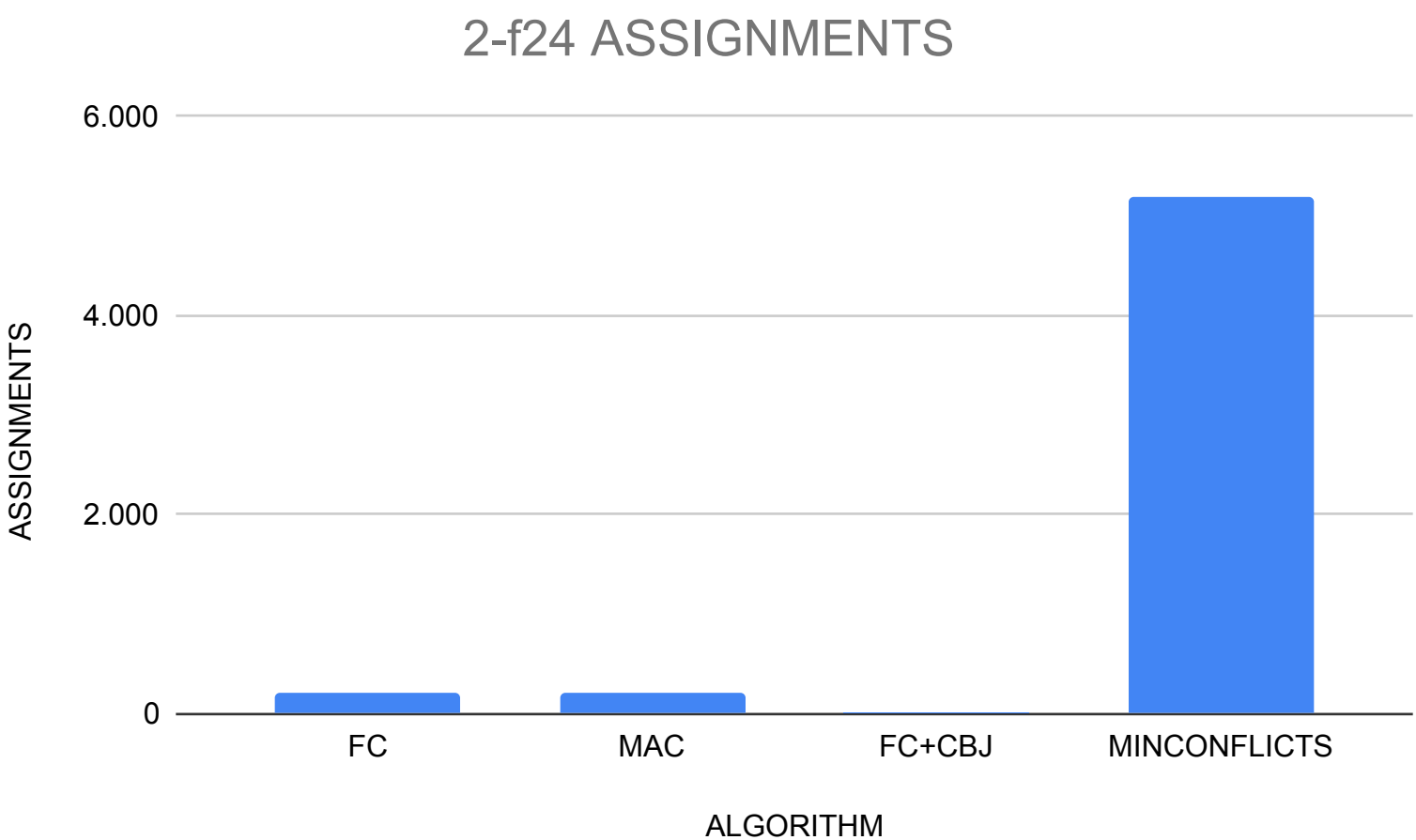
Πρόβλημα 1:

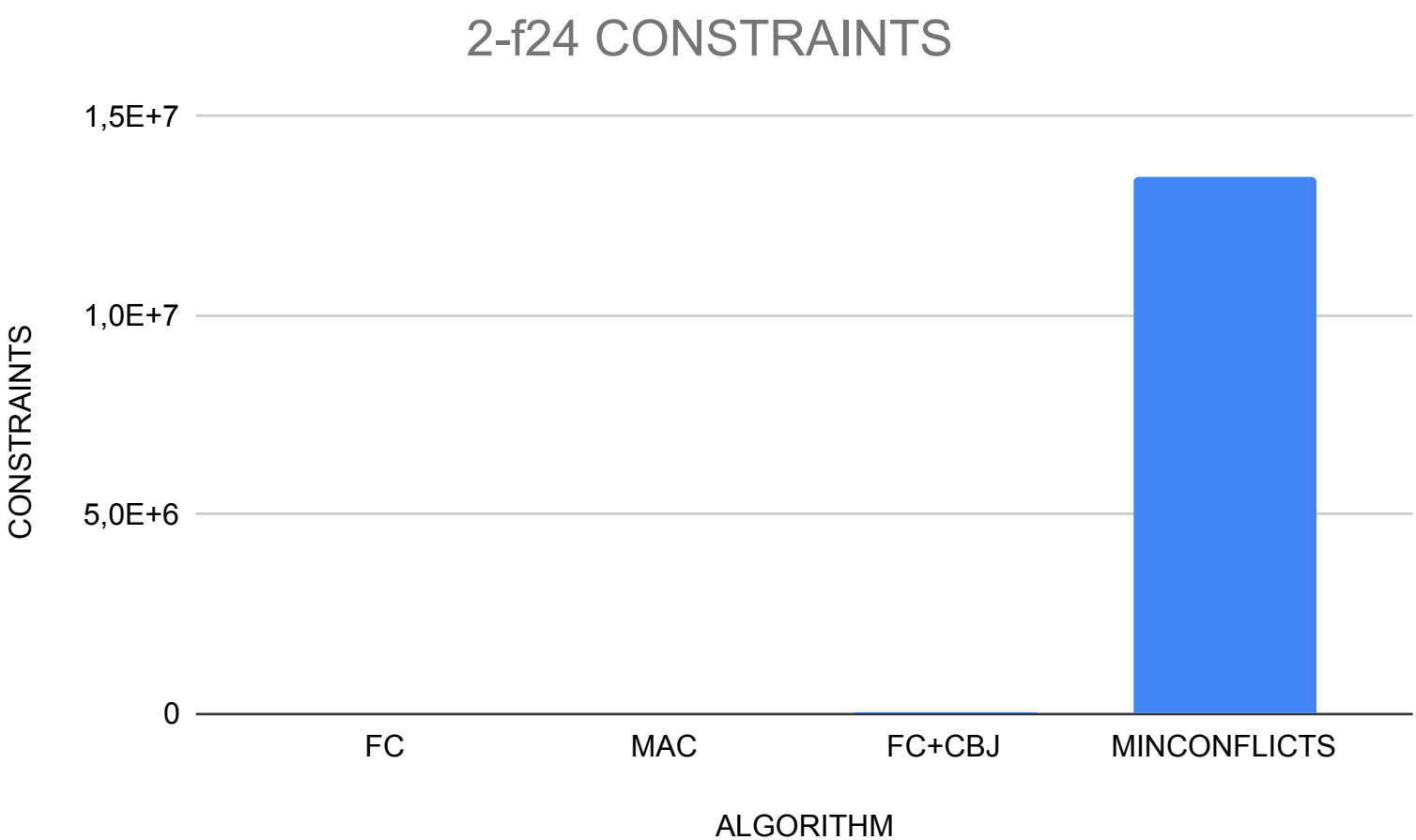
Α) ΜΟΝΤΕΛΟΠΟΙΗΣΗ
Για την μοντελοποίηση του συγκεκριμένου προβλήματος χρησιμοποιήθηκαν τα εξής προκειμένου να αναπαρασταθούν τα δεδομένα του προβλήματος:

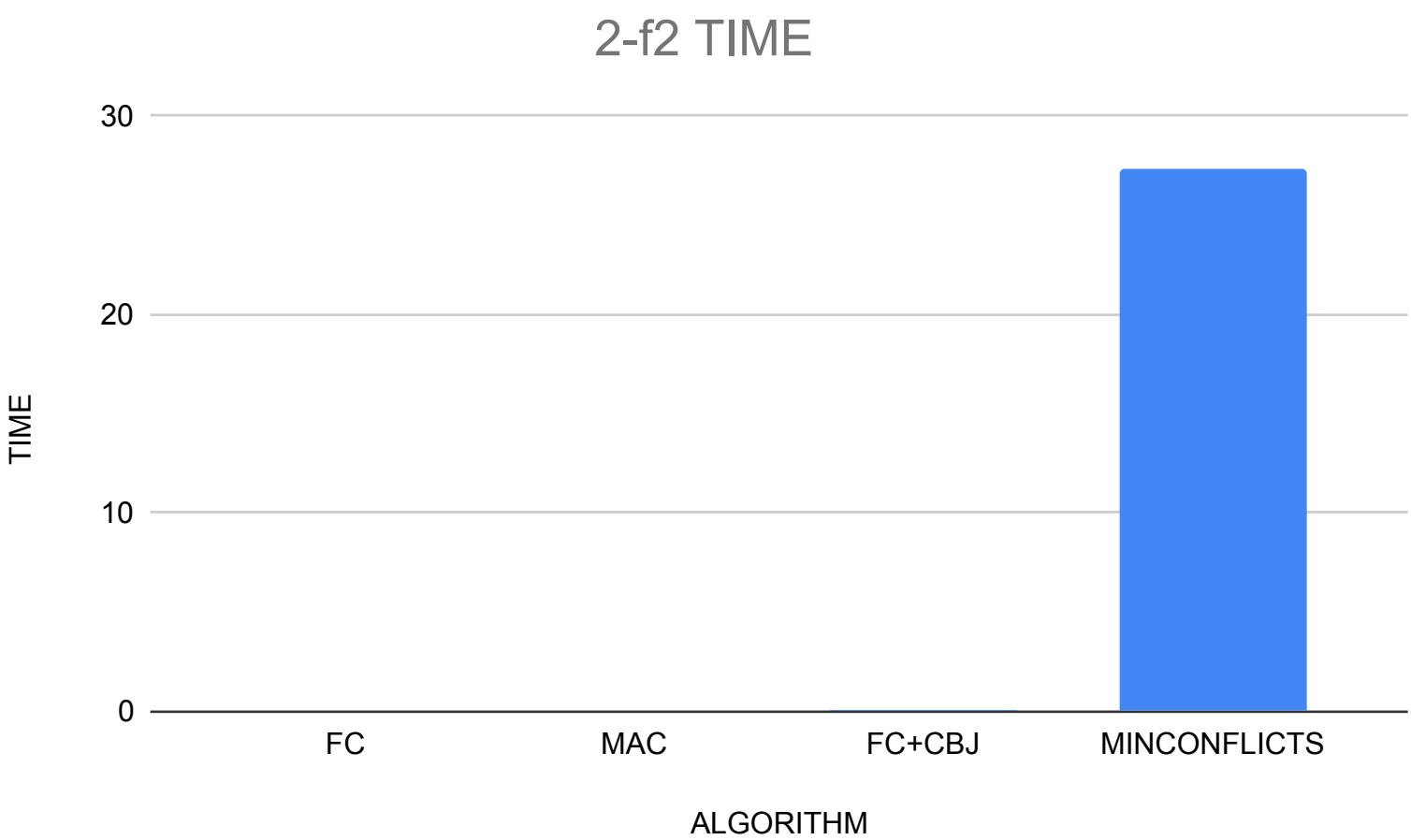
self.variables = list() *// Λίστα με τις μεταβλητές του προβλήματος*
self.domains = dict() *// Λεξικό που αντιστοιχεί κάθε μεταβλητή σε μία λίστα με τις πιθανές τιμές που μπορεί να παρει*
self.op_constraints = dict() *// Λεξικό που αντιστοιχεί κάθε ζεύγος γειτονικών μεταβλητών στο σύμβολο του περιορισμού τους*
self.k_constraints = dict() *// Λεξικό που αντιστοιχεί κάθε ζεύγος γειτονικών μεταβλητών στη σταθερά του περιορισμού τους*
self.neighbours = dict() *// Λεξικό που για κάθε μεταβλητή επιστρέφει μία λίστα με τις μεταβλητές με τις οποίες εμπλέκεται σε περιορισμό*
self.weights = dict() *// Λεξικο που για κάθε ζεύγος μεταβλητών επιστρέφει το βάρος του περιορισμού τους.*

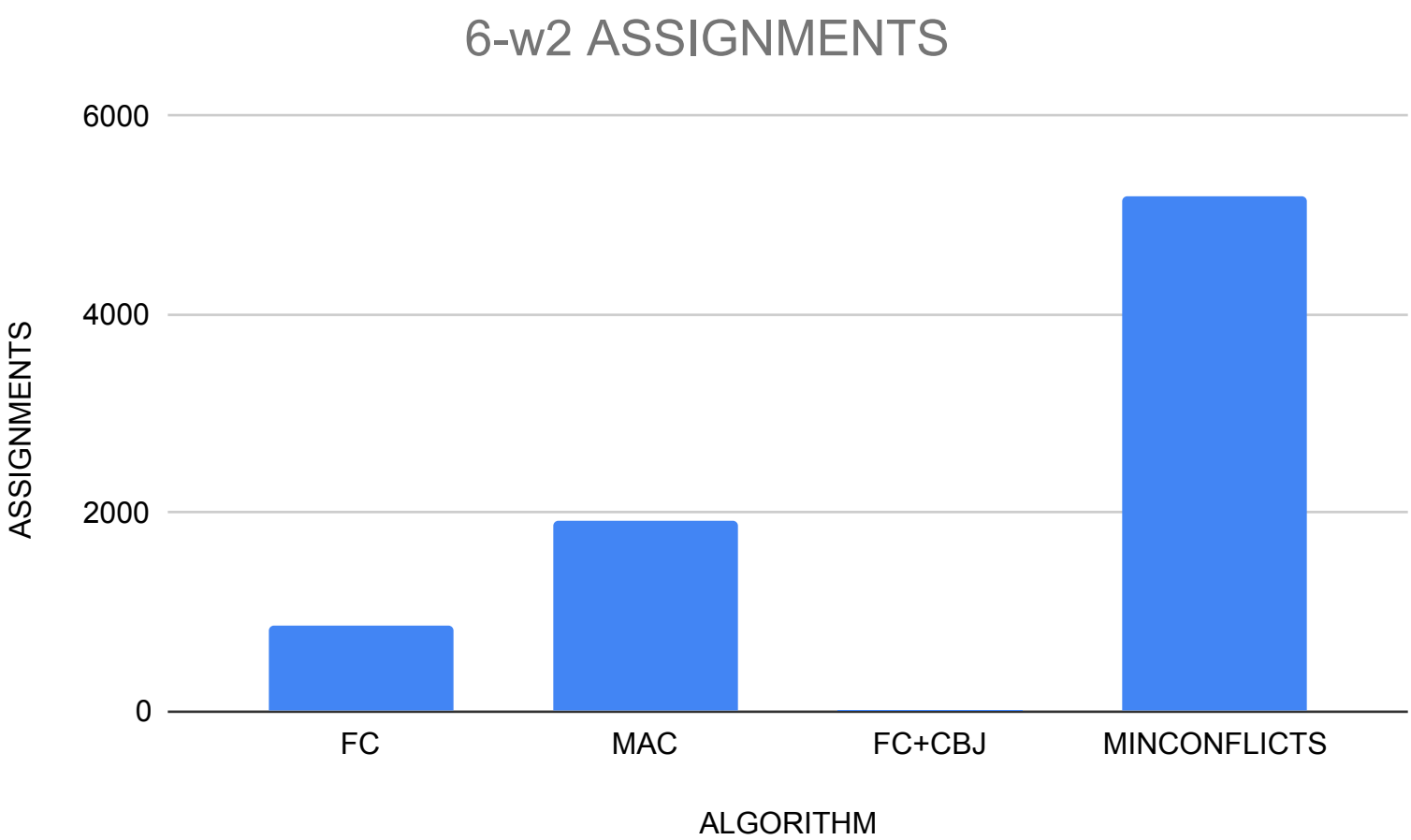
Επίσης έχει υλοποιηθεί η ευρετική που μας δόθηκε και έχει χρησιμοποιηθεί και στους δύο αλγορίθμους FC, MAC, ο FC-CBJ δεν έχει υλοποιηθεί. Ο MINCONFLICTS εκτελείται με μέγιστο αριθμό βημάτων: 5000.

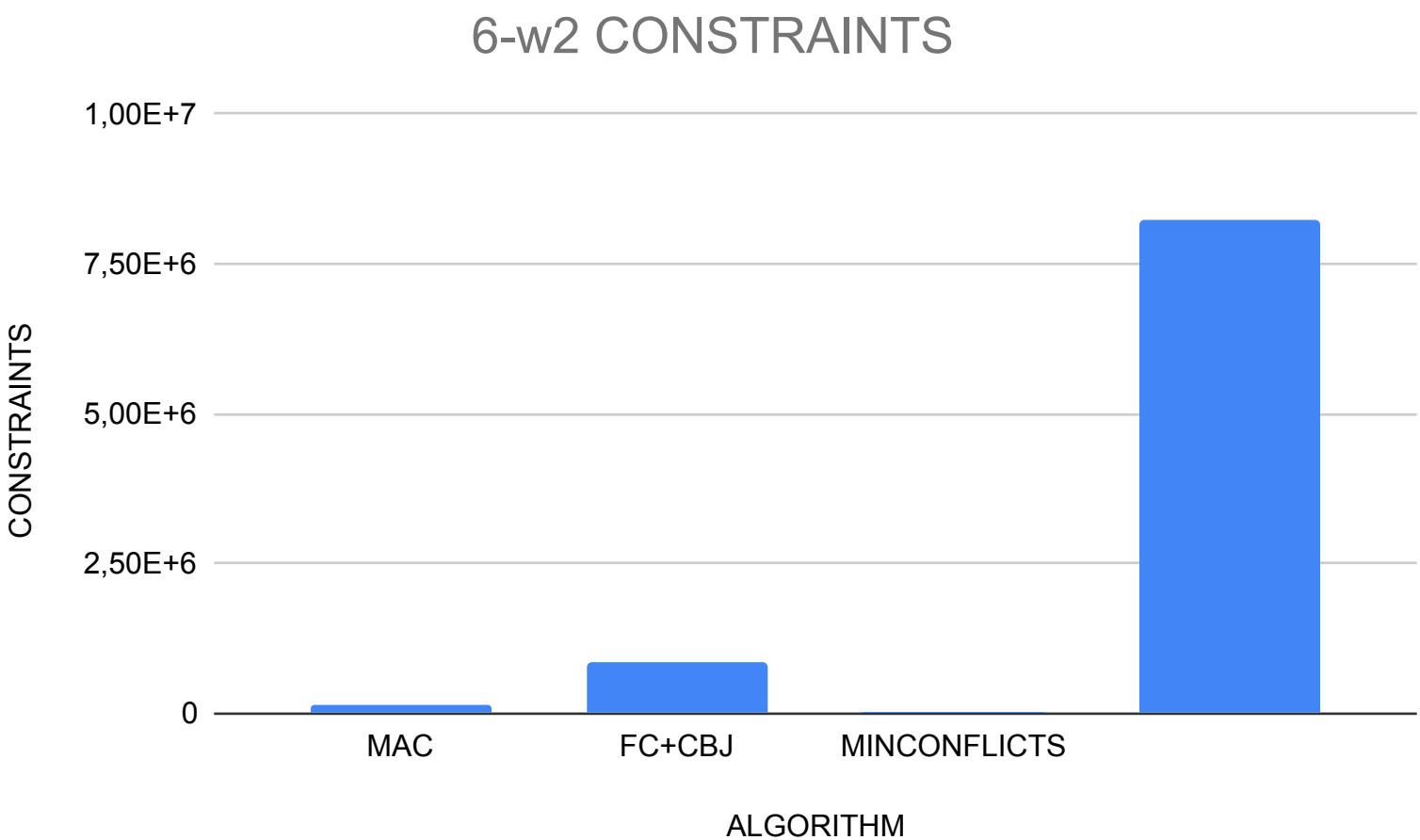
Β) ΑΠΟΤΕΛΕΣΜΑΤΑ
Όσον αφορά τα αποτελέσματα που παραθέτονται παρακάτω παρατηρούμε ότι ο MAC φαίνεται να είναι ο λιγότερο αποδοτικός αλγόριθμος όταν το πρόβλημα είναι μη επιλύσιμο καθώς χρειάζεται τον περισότερο χρόνο από όλους, κάνει τις περισσότερες αναθέσεις τιμών σε μεταβλητές και κάνει τους περισσότερους ελέγχους περιορισμών ακόμα και με την χρήση της ευρετικής συνάρτησης. Αντίθετα όταν το πρόβλημα έχει λύση φαίνεται να είναι πιο αποδοτικός από τον FC. Ακόμα ο FC είτε το πρόβλημα είναι επιλύσιμο είτε όχι βγάζει αποτελέσματα σε φυσιολογικό χρόνο και κάνει λίγες αναθέσεις τιμών και ελέγχους περιορισμών συνεπώς φαίνεται να είναι ο πιο αποδοτικός από όλους. Τέλος ο MINCONFLICTS κάνει πάρα πολύ χρόνο, πολλές αναθέσεις τιμών και πολλούς ελέγχους περιορισμών χωρίς να καταφέρνει τελικά να βρίσκει λύση για το πρόβλημα. Συνεπώς ο MINCONFLICTS κρίνεται πλήρως ακατάλληλος για την επίλυση του συγκεκριμένου προβλήματος.

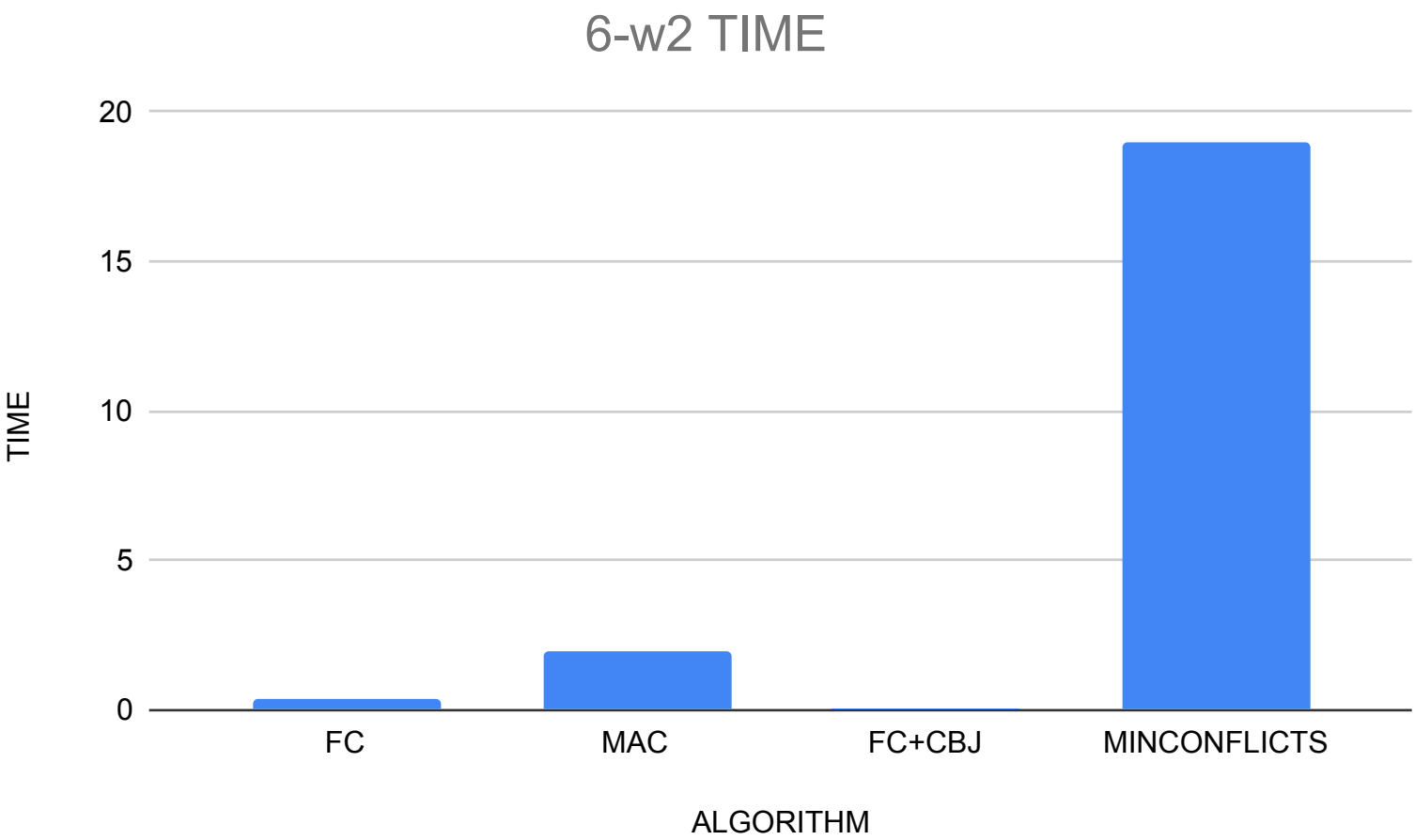

2-f24 ASSIGNMENTS
6.000
4.000
2.000
0
ASSIGNMENTS
FC
MAC
FC+CBJ
MINCONFLICTS
ALGORITHM

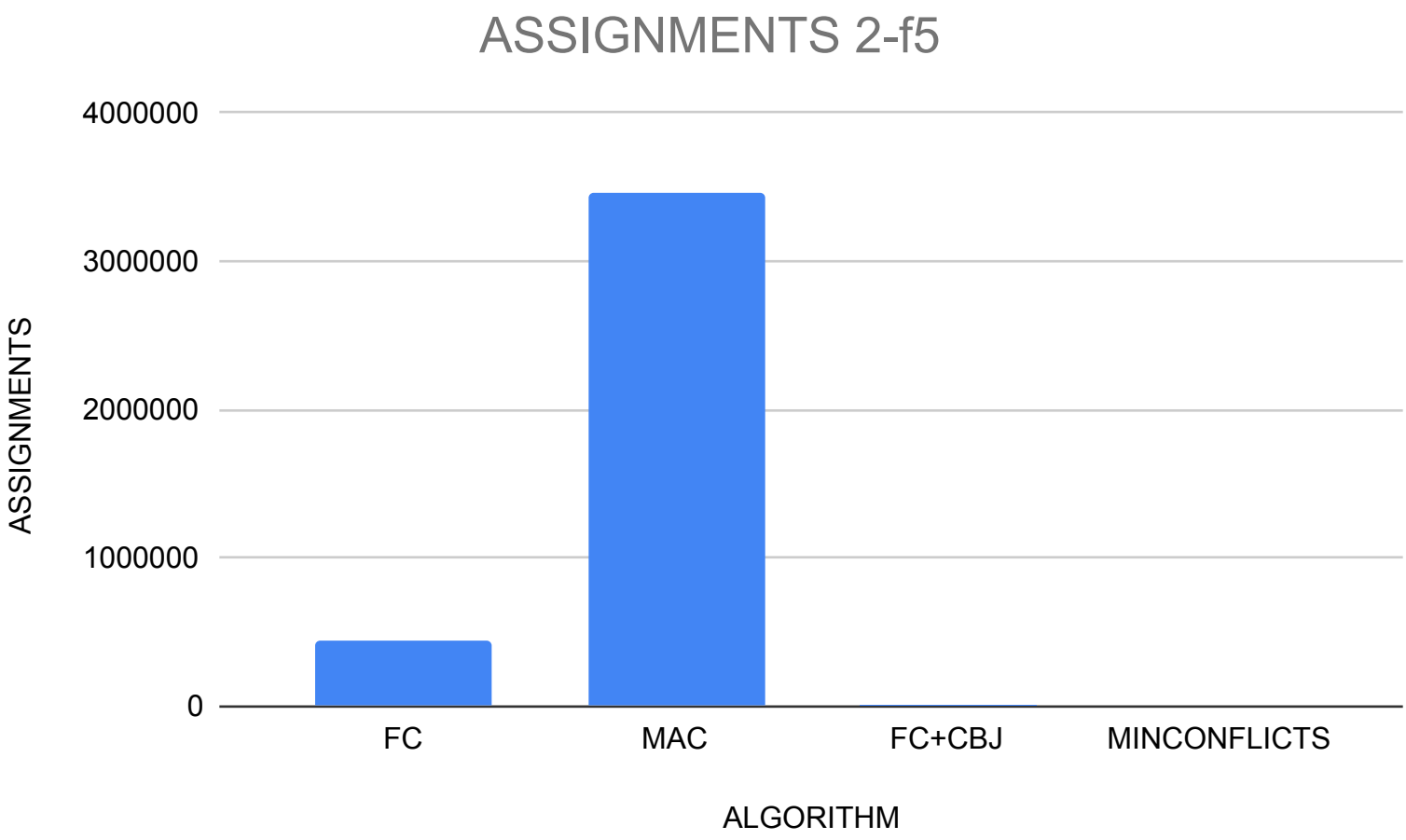

2-f24 CONSTRAINTS
1,5E+7
1,0E+7
5,0E+6
0
CONSTRAINTS
FC
MAC
FC+CBJ
MINCONFLICTS
ALGORITHM

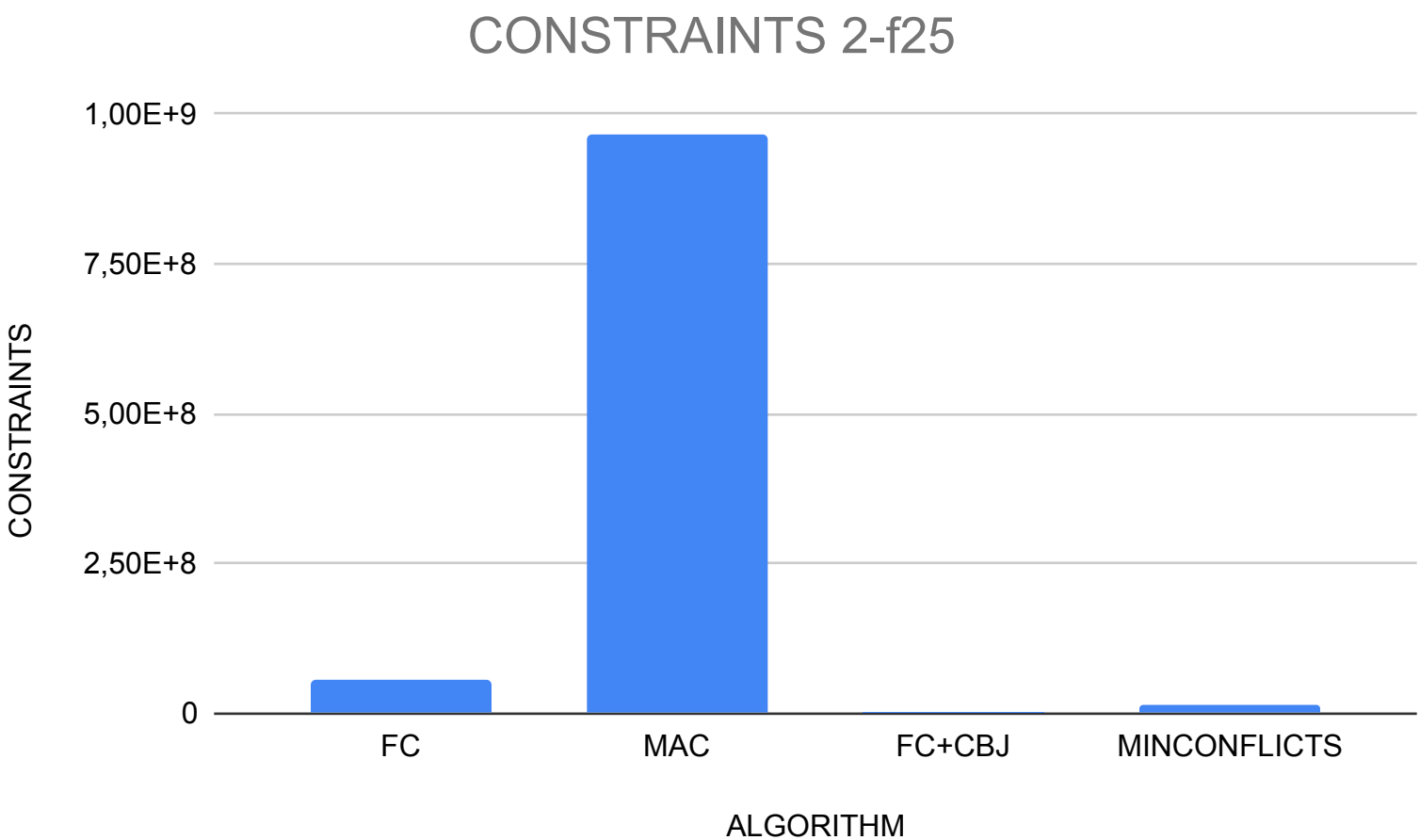

2-f2 TIME
30
20
10
0
TIME
FC
MAC
FC+CBJ
MINCONFLICTS
ALGORITHM

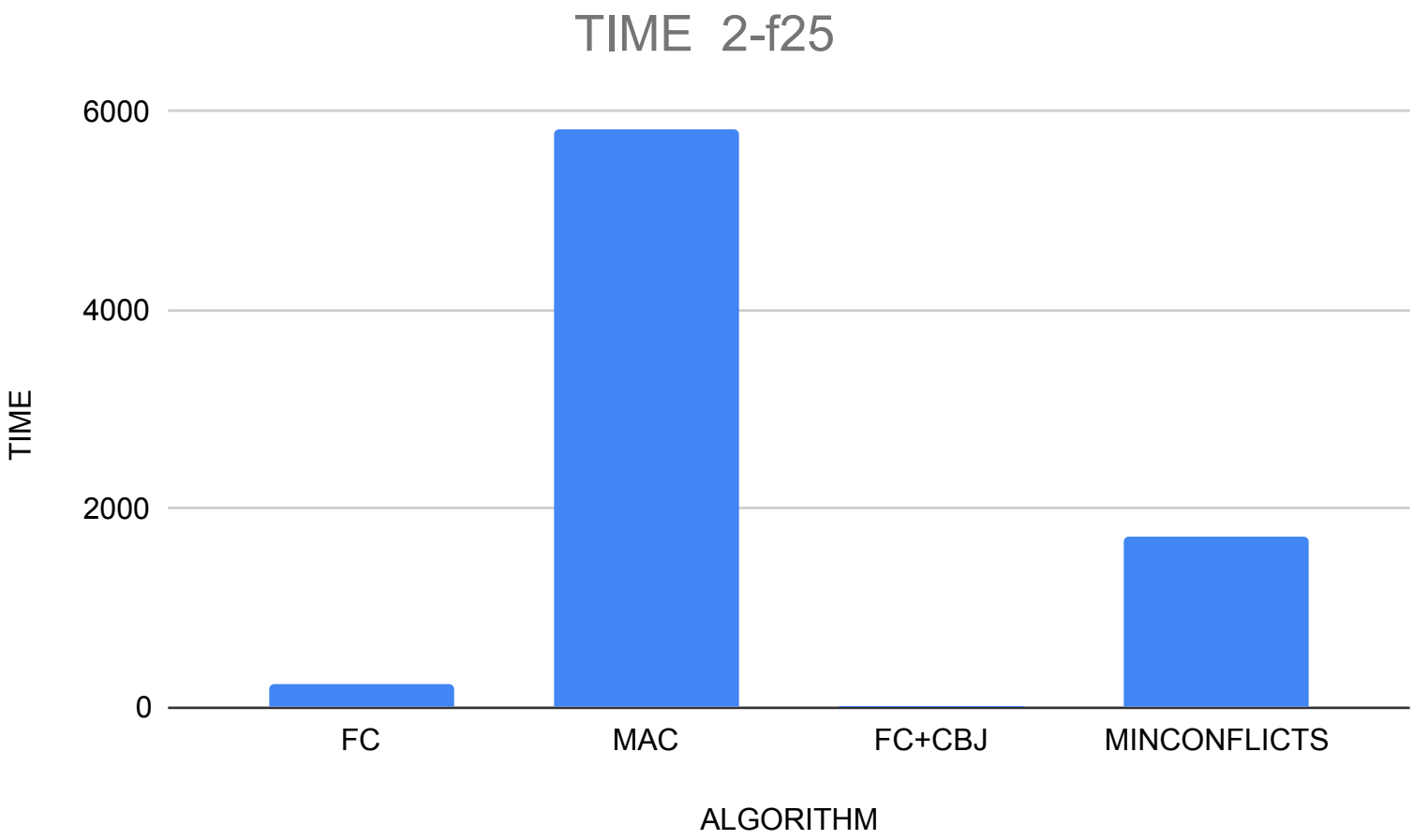

6-w2 ASSIGNMENTS
6000
4000
2000
0
ASSIGNMENTS
FC
MAC
FC+CBJ
MINCONFLICTS
ALGORITHM


6-w2 CONSTRAINTS
1,00E+7
7,50E+6
5,00E+6
2,50E+6
0
CONSTRAINTS
MAC
FC+CBJ
MINCONFLICTS
ALGORITHM


6-w2 TIME
20
15
10
5
0
TIME
FC
MAC
FC+CBJ
MINCONFLICTS
ALGORITHM


ASSIGNMENTS 2-f5
4000000
3000000
2000000
1000000
0
ASSIGNMENTS
FC
MAC
FC+CBJ
MINCONFLICTS
ALGORITHM


CONSTRAINTS 2-f25
1,00E+9
7,50E+8
5,00E+8
2,50E+8
0
CONSTRAINTS
FC
MAC
FC+CBJ
MINCONFLICTS
ALGORITHM


TIME 2-f25
6000
4000
2000
0
TIME
FC
MAC
FC+CBJ
MINCONFLICTS
ALGORITHM

ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ Ι - ΧΕΙΜΕΡΙΝΟ ΕΞΑΜΗΝΟ 2020-2021.

ΕΡΓΑΣΙΑ 3.

ΟΝΟΜΑΤΕΠΩΝΥΜΟ: ΣΚΕΥΟΦΥΛΑΚΑ ΜΑΡΙΑ

Α.Μ: 1115201900173

Πρόβλημα 2.

Ορίζουμε το CSP ως εξής:

Μεταβλητές: {κρεβάτι, γραφείο, καρέκλα, καναπές}.

Domains: $\{x,y \in [0,300] \times [0,400]\}$.

Dκρεβάτι: $\{(x,y): x \in [0,300], y \in [0,400]\}$.

Dγραφείο: $\{(x,y): x \in [0,300], y \in [0,400]\}$.

Dκαρέκλα: $\{(x,y): x \in [0,300], y \in [0,400]\}$.

Dκαναπές: $\{(x,y): x \in [0,300], y \in [0,400]\}$.

Περιορισμοί:

1) Τα έπιπλα δεν πρέπει να εφάπτονται ή να πατάνε το ένα το άλλο.

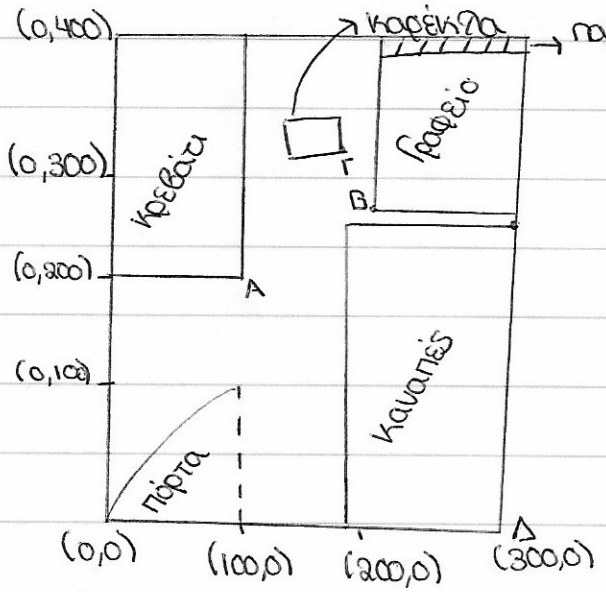
Άρα: $S_i \neq S_j$ και $S_i - S_j > 1$.

2) Γραφείο δίπλα σε είσοδο φωτός.

3) Τίποτα στον χώρο που ανοίγει η πόρτα: $x >, 101$

$y >, 101$.

Οπότε χρησιμοποιώντας τις γνωστές πληροφορίες μία λύση είναι η εξής:



* Κρεβάτι: μήκος: 200, πλάτος: 100, άρα σημείο $A=(100,200)$
* Γραφείο: πλάτος: 160, βάθος: 80, άρα σημείο $B=(220,240)$
* Καναπές: πλάτος: 221, βάθος: 103, άρα σημείο $\Delta=(300,0)$
* Καρέκλα: πλάτος: 41, βάθος: 44, άρα χωράει στο $\Gamma=(200,380)$

∇ Πηγή εισόδου (φωτός) θεωρήθηκε το παράθυρο.

Πρόβλημα 3:

1) Ορίζουμε το CSP:

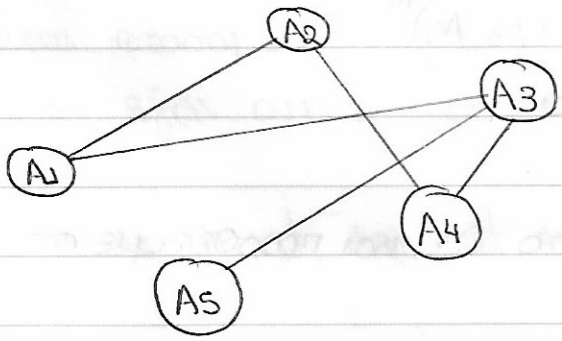
Μεταβλητές: $\{A_1, A_2, A_3, A_4, A_5\}$

Πεδία: $D_1=\{9,10,11\}$, $D_2=\{9,10,11\}$, $D_3=\{9,10,11\}$, $D_4=\{9,10,11\}$, $D_5=\{9,10,11\}$

Περιορισμοί:

* Η A_1 πρέπει να αρχίζει μετά την A_3 → $A_1 > A_3$
* $A_5 < A_3 < A_4$
* $A_2 \neq A_1$ και $A_2 \neq A_4$
* $A_4 \neq 10$

2) Γράφος Περιορισμών:



3) Εφαρμογή AC-3

ορίζουμε:

queue = $\{(A_1,A_3)^1, (A_3,A_5)^2, (A_3,A_4)^3, (A_2,A_1)^4, (A_2,A_4)^5, (A_4,A_4)^6\}$.

domains = $\{D_1, D_2, D_3, D_4, D_5\}$.

Κατά την διάρκεια εκτέλεσης του AC-3 θα αφαιρούνται από τα domains οι τιμές που δεν είναι εφικτές.

$D_1 = \{\clubsuit, \flat, 11\}$ $D_2 = \{9, 10, \natural\}$, $D_3 = \{\clubsuit, 10, 11\}$, $D_4 = \{\clubsuit, \flat, 11\}$.
$D_5 = \{9, \flat, \natural\}$.

Ζωντανά:

1) (A_1,A_3), βγάζουμε το 9 από το D_1 και βάζουμε στο queue τα $(A_3,A_1)^7$ και $(A_2,A_1)^8$

2) (A_3,A_5), βγάζουμε το 9 από το D_3 και βάζουμε στο queue τα $(A_3,A_1)^9$, $(A_3,A_4)^{10}$, $(A_3,A_5)^{11}$.

3) (A_3,A_4), βγάζουμε από το D_3 το 11 και βάζουμε στο queue τα $(A_3,A_1)^{12}$, $(A_3,A_4)^{13}$, $(A_3,A_5)^{14}$

4) (A_2,A_1), δεν αφαιρούμε κάτι

5) (A_2,A_4), -//-

6) (A_4,A_4), βγάζουμε το 10 από το D_4 και προσθέτουμε στο queue το $(A_4,A_4)^{15}$.

7) (A_3,A_1), βγάζουμε το 10 από το D_1 και βάζουμε τα $(A_1,A_3)^{16}$, $(A_5,A_3)^{17}$, $(A_4,A_3)^{18}$ στο queue.

8) (A_2,A_1), βγάζουμε το 11 από το D_2 και προσθέτουμε στο queue τα $(A_1,A_2)^{19}$, $(A_4,A_2)^{20}$

9) (A_3,A_1), δεν αφαιρούμε κάτι.

10) (A_3, A_4), βγάζουμε το 9 από το D_4 και βάζουμε στο ουρά τα $(A_1, A_3)^{21}$, $(A_5, A_3)^{22}$, $(A_4, A_3)^{23}$

11) (A_3, A_5), δεν αφαιρούμε κάτι.

12) (A_3, A_1), —//—

13) (A_3, A_4), —//—

14) (A_3, A_5), —//—

15) (A_4, A_4), —//—

16) (A_1, A_3), —//—

17) (A_5, A_3), βγάζουμε το 11 από το D_5 και προσθέτουμε το $(A_3, A_5)^{24}$ στο ουρά.

18) (A_4, A_3), δεν αφαιρούμε κάτι.

19) (A_1, A_2), —//—

20) (A_4, A_2), —//—

21) (A_1, A_3), —//—

22) (A_5, A_3), —//—

23) (A_4, A_3), —//—

24) (A_3, A_5), —//—

Επομένως φτάσαμε στο σημείο το ουρά να είναι άδειο και έχουμε τα εξής αποτελέσματα:

$A_1 = \{11\}$, $A_2 = \{9, 10\}$, $A_3 = \{10\}$, $A_4 = \{11\}$, $A_5 = \{9\}$.