

ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ
ΕΝΙΣΧΥΤΙΚΗ ΔΙΔΑΣΚΑΛΙΑ 2024-2025

ΧΟΝΔΡΟΜΑΤΙΔΗΣ ΕΛΕΥΘΕΡΙΟΣ

Αλγόριθμοι αντικατάστασης σελίδων

Λίστα αλγορίθμων που θα παρουσιαστούν

- FIFO
- LRU
- OPT

FIFO

FIFO

Στον **FIFO (First In First Out)**, όταν φτάσουμε σε μία σελίδα που δεν υπάρχει στη μνήμη, **αφαιρούμε αυτή που μπήκε πιο παλιά στη μνήμη** και στη θέση της βάζουμε τη νέα σελίδα. Σε τέτοιες περιπτώσεις, λέμε ότι έχουμε **σφάλμα σελίδας!**

Επίσης, αν η σελίδα η οποία εξετάζουμε **υπάρχει ήδη** στη μνήμη, δεν πειράζουμε τίποτα και απλά προχωράμε στην επόμενη.

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης FIFO;

1 6 7 9 1 9 2 3 2 1

ΣΦΑΛΜΑ?

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης FIFO;

1 6 7 9 1 9 2 3 2 1

ΣΦΑΛΜΑ?

Σαν πρώτο βήμα, σε ΟΛΟΥΣ τους αλγορίθμους αντικατάστασης, **όποιοι και να έχουμε**, γεμίζουμε τη μνήμη. Άρα, θα βάλουμε κατευθείαν μέσα την 1, την 6, την 7 και την 9. Επίσης, αφού η μνήμη δεν τις είχε όντας άδεια, δηλώνουμε πως έχουμε σφάλμα σελίδας. **ΠΡΟΣΟΧΗ!!! Βάζουμε τις σελίδες μία προς μία, με τη σειρά που εμφανίζονται.**

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης FIFO;

1 6 7 9 1 9 2 3 2 1

1									
ΣΦΑΛΜΑ? X									

ΣΦΑΛΜΑ?

Σαν πρώτο βήμα, σε ΟΛΟΥΣ τους αλγορίθμους αντικατάστασης, **όποιοι και να έχουμε** γεμίζουμε τη μνήμη. Άρα, θα βάλουμε κατευθείαν μέσα την 1, την 6, την 7 και την 9. Επίσης, αφού η μνήμη δεν τις είχε όντας άδεια, δηλώνουμε πως έχουμε σφάλμα σελίδας. **ΠΡΟΣΟΧΗ!!! Βάζουμε τις σελίδες μία προς μία, με τη σειρά που εμφανίζονται.**

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης FIFO;

1 6 7 9 1 9 2 3 2 1

1	1								
	6								
ΣΦΑΛΜΑ?	X	X							

Σαν πρώτο βήμα, σε ΟΛΟΥΣ τους αλγορίθμους αντικατάστασης, **όποιοι και να έχουμε** γεμίζουμε τη μνήμη. Άρα, θα βάλουμε κατευθείαν μέσα την 1, την 6, την 7 και την 9. Επίσης, αφού η μνήμη δεν τις είχε όντας άδεια, δηλώνουμε πως έχουμε σφάλμα σελίδας. **ΠΡΟΣΟΧΗ!!! Βάζουμε τις σελίδες μία προς μία, με τη σειρά που εμφανίζονται.**

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης FIFO;

1 6 7 9 1 9 2 3 2 1

1	1	1							
	6	6							
		7							
ΣΦΑΛΜΑ?	X	X	X						

Σαν πρώτο βήμα, σε ΟΛΟΥΣ τους αλγορίθμους αντικατάστασης, **όποιοι και να έχουμε** γεμίζουμε τη μνήμη. Άρα, θα βάλουμε κατευθείαν μέσα την 1, την 6, την 7 και την 9. Επίσης, αφού η μνήμη δεν τις είχε όντας άδεια, δηλώνουμε πως έχουμε σφάλμα σελίδας. **ΠΡΟΣΟΧΗ!!! Βάζουμε τις σελίδες μία προς μία, με τη σειρά που εμφανίζονται.**

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης FIFO;

1 6 7 9 1 9 2 3 2 1

1	1	1	1						
	6	6	6						
		7	7						
			9						
ΣΦΑΛΜΑ?	X	X	X						

Έχοντας πλέον γεμίσει πλήρως τη μνήμη, από εδώ και στο εξής, θα ελέγχουμε αν οι σελίδες που βλέπουμε από πάνω, υπάρχουν ή όχι σε κάποια από τις θέσεις της μνήμης. Στο παραπάνω παράδειγμα, η σελίδα 1 υπάρχει ήδη στη μνήμη. Επομένως, **ΔΕΝ ΕΧΟΥΜΕ ΣΦΑΛΜΑ ΣΕΛΙΔΑΣ**, και απλά προχωράμε.

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης FIFO;

1 6 7 9 1 9 2 3 2 1

1	1	1	1	1					
	6	6	6	6					
		7	7	7					
			9	9					
ΣΦΑΛΜΑ?	X	X	X	X					

Συνεχίζοντας, η σελίδα 9 υπάρχει επίσης στη μνήμη. Άρα, ΔΕΝ ΕΧΟΥΜΕ ΣΦΑΛΜΑ ΣΕΛΙΔΑΣ!

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης FIFO;

1 6 7 9 1 9 2 3 2 1

1	1	1	1	1	1				
	6	6	6	6	6				
		7	7	7	7				
			9	9	9				
ΣΦΑΛΜΑ?	X	X	X	X					

Φτάνουμε στη σελίδα 2 αλλά όπως μπορούμε να διακρίνουμε, αυτή η σελίδα ΔΕΝ ΥΠΑΡΧΕΙ στη μνήμη. Συνεπώς, πρέπει να κοιτάξουμε πότε κάθε σελίδα μπήκε στη μνήμη και να εντοπίσουμε τη παλιότερη.

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης FIFO;

	1	6	7	9	1	9	2	3	2	1
Εδώ μπήκε η 1	1	1	1	1	1	1				
Εδώ μπήκε η 6		6	6	6	6	6				
Εδώ μπήκε η 7			7	7	7	7				
Εδώ μπήκε η 9				9	9	9				
ΣΦΑΛΜΑ?	X	X	X	X						

Όπως βλέπουμε, η πιο παλιά σελίδα είναι η 1. Άρα, θα την αφαιρέσουμε από τη μνήμη, και στη θέση της θα βάλουμε τη 2. Επιπλέον, αφού η 2 δεν υπήρχε στη μνήμη σημαίνει ότι **ΕΧΟΥΜΕ ΣΦΑΛΜΑ ΜΝΗΜΗΣ**.

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης FIFO;

1 6 7 9 1 9 2 3 2 1

1	1	1	1	1	1	<u>2</u>			
	6	6	6	6	6	6			
		7	7	7	7	7			
			9	9	9	9			
ΣΦΑΛΜΑ?	X	X	X	X		X			

Η σελίδα 3, επίσης δεν υπάρχει μέσα στη μνήμη. Άρα, **ΕΧΟΥΜΕ ΣΦΑΛΜΑ ΣΕΛΙΔΑΣ** και πρέπει να ελέγξουμε ξανά ποια είναι η πιο παλιά σελίδα. ΠΡΟΣΟΧΗ!!! Η σελίδα 1 έχει αντικατασταθεί, άρα την εξαιρούμε από αυτή τη διαδικασία.

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης FIFO;

	1	6	7	9	1	9	2	3	2	1
Εδώ μπήκε η 2	1	1	1	1	1	1	2			
Εδώ μπήκε η 6		6	6	6	6	6	6			
Εδώ μπήκε η 7			7	7	7	7	7			
Εδώ μπήκε η 9				9	9	9	9			
ΣΦΑΛΜΑ?	X	X	X	X			X			

Παρατηρούμε πως η πιο πρόσφατη σελίδα που μπήκε στη μνήμη είναι η 2 (που βάλαμε μόλις πριν λίγο), και η πιο παλιά σελίδα είναι η 6. Επομένως, θα αφαιρέσουμε την 6 και στη θέση της θα μπει η σελίδα 3.

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης FIFO;

1 6 7 9 1 9 2 3 2 1

1	1	1	1	1	1	2	2		
	6	6	6	6	6	6	<u>3</u>		
		7	7	7	7	7	7		
			9	9	9	9	9		
ΣΦΑΛΜΑ?	X	X	X	X		X	X		

Στη συνέχεια, συναντάμε τη σελίδα 2. Η σελίδα αυτή όμως υπάρχει ήδη μέσα στη μνήμη, άρα **ΔΕΝ ΕΧΟΥΜΕ ΣΦΑΛΜΑ ΣΕΛΙΔΑΣ** και απλά προχωράμε στην επόμενη.

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης FIFO;

1 6 7 9 1 9 2 3 2 1

1	1	1	1	1	1	2	2	2	
	6	6	6	6	6	6	3	3	
		7	7	7	7	7	7	7	
			9	9	9	9	9	9	
ΣΦΑΛΜΑ?	X	X	X	X		X	X		

Τέλος, έχουμε τη σελίδα 1. Η σελίδα αυτή **δεν υπάρχει μέσα στη μνήμη αυτή τη στιγμή**. Επομένως, **ΕΧΟΥΜΕ ΣΦΑΛΜΑ ΣΕΛΙΔΑΣ** και πρέπει να αναζητήσουμε για τη σελίδα που μπήκε πιο παλιά από όλες στη μνήμη.

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης FIFO;

	1	6	7	9	1	9	2	3	2	1
Εδώ μπήκε η 2							2	2	2	
Εδώ μπήκε η 3	1	1	1	1	1	1	2	2	2	
		6	6	6	6	6	6	3	3	
Εδώ μπήκε η 7				7	7	7	7	7	7	
Εδώ μπήκε η 9					9	9	9	9	9	
ΣΦΑΛΜΑ?	X	X	X	X			X	X		

Όπως φαίνεται, η πιο παλιά σελίδα είναι η 7. Επομένως, αυτή θα αφαιρεθεί από τη μνήμη και στη θέση της θα μπει η σελίδα 1.

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης FIFO;

1 6 7 9 1 9 2 3 2 1

1	1	1	1	1	1	2	2	2	2
	6	6	6	6	6	6	3	3	3
		7	7	7	7	7	7	7	<u>1</u>
			9	9	9	9	9	9	9
X	X	X	X			X	X		X

ΣΦΑΛΜΑ?

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης FIFO;

1 6 7 9 1 9 2 3 2 1

1	1	1	1	1	1	2	2	2	2
	6	6	6	6	6	6	3	3	3
		7	7	7	7	7	7	7	1
			9	9	9	9	9	9	9
ΣΦΑΛΜΑ?	X	X	X	X		X	X		X

Επομένως, έχοντας φτάσει στο τέλος είχαμε συνολικά 7 σφάλματα σελίδας.

Άσκηση εξάσκησης με τον αλγόριθμο FIFO!

Για την ακολουθία αναφορών {1,2,3,4,5,1,3,1,6,3,2,3}, υπολογίστε πόσα σφάλματα μνήμης θα έχουμε αν χρησιμοποιήσουμε τον αλγόριθμο αντικατάστασης FIFO με 4 πλαίσια.

ΛΥΣΗ

Για την ακολουθία αναφορών {1,2,3,4,5,1,3,1,6,3,2,3}, υπολογίστε πόσα σφάλματα σελίδας θα έχουμε αν χρησιμοποιήσουμε τον αλγόριθμο αντικατάστασης FIFO με 4 πλαίσια.

1	2	3	4	5	1	3	1	6	3	2	3
---	---	---	---	---	---	---	---	---	---	---	---

1	1	1	1	5	5	5	5	5	5	2	2
	2	2	2	2	1	1	1	1	1	1	1
		3	3	3	3	3	3	6	6	6	6
			4	4	4	4	4	4	3	3	3

ΣΦΑΛΜΑ?

X	X	X	X	X	X			X	X	X	
---	---	---	---	---	---	--	--	---	---	---	--

Είχαμε συνολικά 9 σφάλματα σελίδας!

LRU

LRU

Στον αλγόριθμο **LRU (Least Recently Used)**, όταν φτάσουμε σε μία σελίδα η οποία δεν βρίσκεται στη μνήμη, **πρέπει να κοιτάξουμε πότε ήταν η τελευταία φορά που συναντήσαμε κάθε μία από τις σελίδες που βρίσκονται στη μνήμη**. Μόλις εντοπίσουμε αυτή που συναντήσαμε για τελευταία φορά παλαιότερα από όλες (δηλαδή τη πιο αριστερή), την αφαιρούμε και βάζουμε στη θέση της τη νέα σελίδα που συναντήσαμε.

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης LRU;

2 4 6 2 7 5 1 3 6 4

ΣΦΑΛΜΑ?

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης LRU;

2 4 6 2 7 5 1 3 6 4

ΣΦΑΛΜΑ?

Καταρχάς, όπως και στον FIFO, πρέπει να γεμίσουμε όλες τις θέσεις της μνήμης. Υπενθυμίζεται ότι αφού η μνήμη είναι άδεια, **ό,τι και να βάλουμε θα προκαλέσει σφάλμα σελίδας.**

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης LRU;

2 4 6 2 7 5 1 3 6 4

<u>2</u>									
ΣΦΑΛΜΑ? X									

Καταρχάς, όπως και στον FIFO, καλούμαστε να γεμίσουμε όλες τις θέσεις της μνήμης. Υπενθυμίζεται ότι αφού η μνήμη είναι άδεια, **ό,τι και να βάλουμε θα προκαλέσει σφάλμα σελίδας.**

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης LRU;

2 4 6 2 7 5 1 3 6 4

2	2								
	<u>4</u>								
ΣΦΑΛΜΑ?	X	X							

Καταρχάς, όπως και στον FIFO, καλούμαστε να γεμίσουμε όλες τις θέσεις της μνήμης. Υπενθυμίζεται ότι αφού η μνήμη είναι άδεια, ό,τι και να βάλουμε θα προκαλέσει σφάλμα σελίδας.

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης LRU;

2 4 6 2 7 5 1 3 6 4

2	2	2							
	4	4							
		<u>6</u>							
ΣΦΑΛΜΑ?	X	X							

Αφού γεμίσει η μνήμη, μπορούμε να συνεχίσουμε με τη σελίδα 2. Αυτή, υπάρχει ήδη μέσα στη μνήμη, άρα δεν έχουμε σφάλμα σελίδας και δεν αλλάζουμε τίποτα.

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης LRU;

2 4 6 2 7 5 1 3 6 4

2	2	2	2						
	4	4	4						
		6	6						
ΣΦΑΛΜΑ?	X	X	X						

Η σελίδα 7 όμως, δεν υπάρχει πουθενά στη μνήμη! Άρα, **έχουμε σφάλμα σελίδας** και θα πρέπει να ελέγξουμε πότε ήταν η τελευταία φορά που συναντήσαμε τη 2, την 6 και τη 4 σελίδα κατά την εκτέλεση του αλγορίθμου.

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης LRU;

Τελευταία φορά που συναντήσαμε τη 4 Τελευταία φορά που συναντήσαμε την 6 Τελευταία φορά που συναντήσαμε την 2



2	2	2	2						
	4	4	4						
		6	6						
ΣΦΑΛΜΑ?	X	X	X						

Όπως βλέπουμε, η σελίδα 4 είναι αυτή που συναντήσαμε για τελευταία φορά παλαιότερα (το καταλαβαίνουμε βλέποντας ποια από τις σελίδες είναι η πιο αριστερή). Άρα, την αφαιρούμε από τη μνήμη και την αντικαθιστούμε με την σελίδα 7.

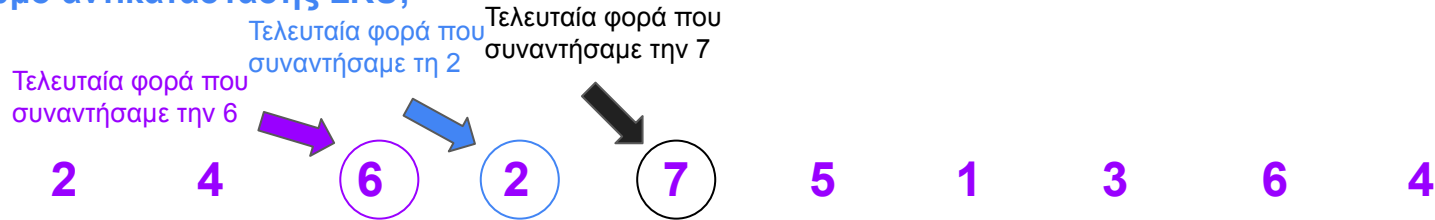
Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης LRU;

2 4 6 2 7 5 1 3 6 4

2	2	2	2	2					
	4	4	4	<u>7</u>					
		6	6	6					
ΣΦΑΛΜΑ?	X	X	X	X					

Συνεχίζοντας, η σελίδα 5 δεν υπάρχει στη μνήμη! Επομένως, έχουμε για άλλη μια φορά σφάλμα σελίδας και πρέπει να εντοπίσουμε πότε ήταν η τελευταία φορά που συναντήσαμε τη 2, την 7 και την 6 σελίδα κατά την εκτέλεση του αλγορίθμου.

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης LRU;



2	2	2	2	2					
	4	4	4	7					
		6	6	6					
ΣΦΑΛΜΑ?	X	X		X					

Όπως βλέπουμε, η σελίδα 6 είναι αυτή που συναντήσαμε για τελευταία φορά παλαιότερα. Άρα, την αφαιρούμε και βάζουμε στη θέση της τη σελίδα 5.

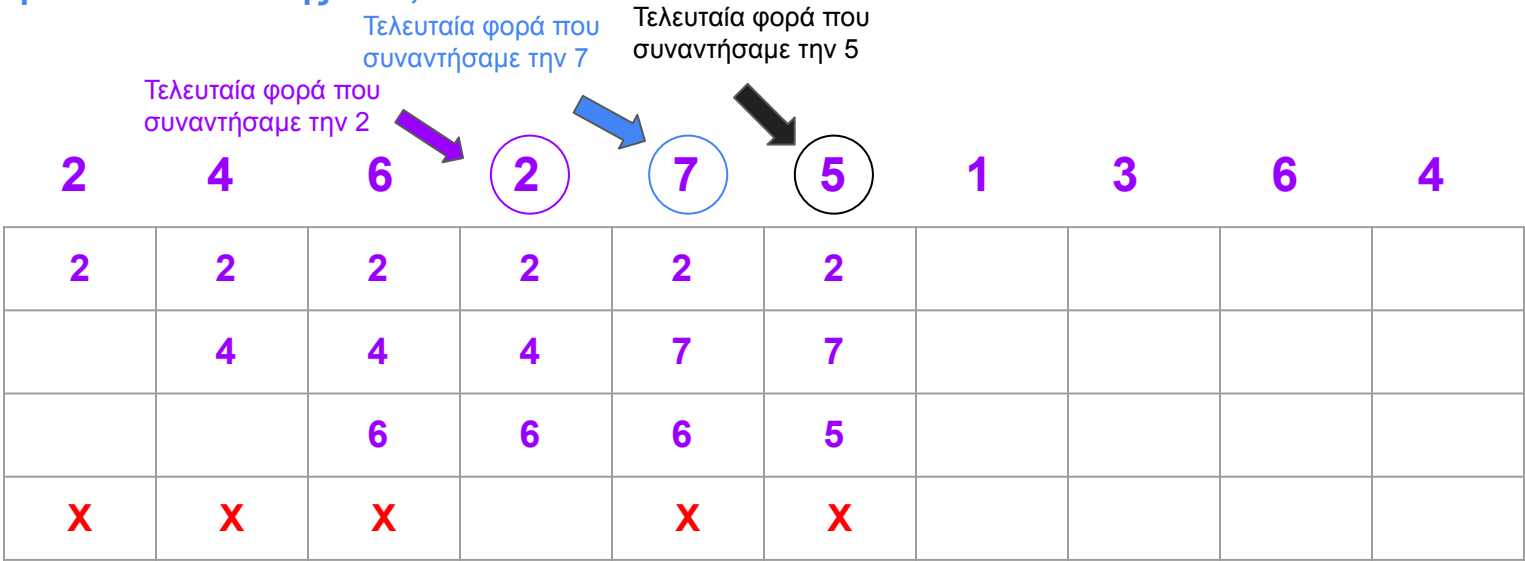
Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης LRU;

2 4 6 2 7 5 1 3 6 4

2	2	2	2	2	2				
	4	4	4	7	7				
		6	6	6	<u>5</u>				
ΣΦΑΛΜΑ?	X	X	X	X	X				

Στη συνέχεια, η σελίδα 1 δεν βρίσκεται στη μνήμη! Άρα, έχουμε ξανά σφάλμα σελίδας και πρέπει να εντοπίσουμε πότε ήταν η τελευταία φορά που συναντήσαμε τη 2, την 7 και την 5 σελίδα κατά την εκτέλεση του αλγορίθμου.

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης LRU;



Όπως βλέπουμε, η σελίδα 2 είναι αυτή που συναντήσαμε για τελευταία φορά παλαιότερα. Άρα, την αφαιρούμε και βάζουμε στη θέση της τη σελίδα 1.

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης LRU;

2 4 6 2 7 5 1 3 6 4

2	2	2	2	2	2	<u>1</u>			
	4	4	4	7	7	7			
		6	6	6	5	5			
ΣΦΑΛΜΑ?	X	X	X	X	X	X			

Προχωρώντας, η σελίδα 3 δεν υπάρχει στη μνήμη! Επομένως, **έχουμε ξανά σφάλμα σελίδας** και θα ελέγξουμε πότε ήταν η τελευταία φορά που συναντήσαμε την 1, την 7 και την 5 σελίδα κατά την εκτέλεση του αλγορίθμου.

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης LRU;



Όπως βλέπουμε, η σελίδα 7 είναι αυτή που συναντήσαμε για τελευταία φορά παλαιότερα. Άρα, την αφαιρούμε και βάζουμε στη θέση της τη σελίδα 3.

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης LRU;

2 4 6 2 7 5 1 3 6 4

2	2	2	2	2	2	1	1		
	4	4	4	7	7	7	<u>3</u>		
		6	6	6	5	5	5		
ΣΦΑΛΜΑ?	X	X	X		X	X	X	X	

Επαναλαμβάνουμε την όλη διαδικασία άλλες 2 φορές μέχρι να φτάσουμε στο τέλος.

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης LRU;

2 4 6 2 7 5 1 3 6 4

2	2	2	2	2	2	1	1	1	4
	4	4	4	7	7	7	3	3	3
		6	6	6	5	5	5	6	6
ΣΦΑΛΜΑ?	X	X		X	X	X	X	X	X

Συνολικά, είχαμε 9 σφάλματα σελίδας!

Άσκηση εξάσκησης με τον αλγόριθμο LRU!

Για την ακολουθία αναφορών $\{7,0,1,2,0,3,0,4,2,3\}$, υπολογίστε πόσα σφάλματα μνήμης θα έχουμε αν χρησιμοποιήσουμε τον αλγόριθμο αντικατάστασης LRU με 3 πλαίσια.

ΛΥΣΗ

Για την ακολουθία αναφορών {7,0,1,2,0,3,0,4,2,3}, υπολογίστε πόσα σφάλματα μνήμης θα έχουμε αν χρησιμοποιήσουμε τον αλγόριθμο αντικατάστασης LRU με 3 πλαίσια.

7 0 1 2 0 3 0 4 2 3

7	7	7	2	2	2	2	4	4	4
	0	0	0	0	0	0	0	0	3
		1	1	1	3	3	3	2	2
ΣΦΑΛΜΑ?	X	X	X	X		X		X	X

Συνολικά, είχαμε 8 σφάλματα σελίδας!

ΠΡΟΣΟΧΗ!!!

Δεν κοιτάμε μόνο τις 3-4 τελευταίες σελίδες όπως ενδέχεται να σκεφτήκατε βάσει του προηγούμενου παραδείγματος. Κοιτάμε πότε συναντήσαμε **ΤΕΛΕΥΤΑΙΑ ΦΟΡΑ** τη κάθε σελίδα που υπάρχει μέσα στη μνήμη.

Τελευταία φορά που συναντήσαμε τη 2

7 0 1 2 0 3 0 4 2 3

7	7	7	2	2	2	2	4	4	4
	0	0	0	0	0	0	0	0	3
		1	1	1	3	3	3	2	2
ΣΦΑΛΜΑ?	X	X	X	X		X		X	X

OPT

OPT

Στον **OPT**, όταν φτάσουμε σε μία σελίδα που δεν υπάρχει στη μνήμη, με βάση αυτές που έχουμε μέσα της, **αφαιρούμε εκείνη που δε θα συναντήσουμε ξανά για το μεγαλύτερο διάστημα στο μέλλον** και στη θέση της βάζουμε τη νέα σελίδα. Σε τέτοιες περιπτώσεις, λέμε ότι έχουμε **σφάλμα σελίδας!**

Αν μια σελίδα δεν θα τη συναντήσουμε ποτέ ξανά, την αφαιρούμε, και στη θέση της βάζουμε τη νέα σελίδα.

Αν πολλαπλές σελίδες δε τις συναντήσουμε ξανά, επιλέγουμε τυχαία ποια θα αφαιρέσουμε.

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης OPT;

2 4 6 1 4 5 6 2 1 6

ΣΦΑΛΜΑ?

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης OPT;

2 4 6 1 4 5 6 2 1 6

ΣΦΑΛΜΑ?

Ξεκινώντας, όπως στον FIFO και στον LRU, πρέπει να γεμίσουμε όλες τις θέσεις της μνήμης. Υπενθυμίζεται ότι αφού η μνήμη είναι άδεια, **ό,τι και να βάλουμε θα προκαλέσει σφάλμα σελίδας.**

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης OPT;

2 4 6 1 4 5 6 2 1 6

<u>2</u>									
ΣΦΑΛΜΑ? X									

Ξεκινώντας, όπως στον FIFO και στον LRU, πρέπει να γεμίσουμε όλες τις θέσεις της μνήμης. Υπενθυμίζεται ότι αφού η μνήμη είναι άδεια, **ό,τι και να βάλουμε θα προκαλέσει σφάλμα σελίδας.**

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης OPT;

2 4 6 1 4 5 6 2 1 6

2	2								
	<u>4</u>								
ΣΦΑΛΜΑ?	X	X							

Ξεκινώντας, όπως στον FIFO και στον LRU, πρέπει να γεμίσουμε όλες τις θέσεις της μνήμης. Υπενθυμίζεται ότι αφού η μνήμη είναι άδεια, **ό,τι και να βάλουμε θα προκαλέσει σφάλμα σελίδας.**

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης OPT;

2 4 6 1 4 5 6 2 1 6

2	2	2							
	4	4							
		<u>6</u>							
ΣΦΑΛΜΑ?	X	X	X						

Έχοντας γεμίσει τη μνήμη, μπορούμε να συνεχίσουμε με τη σελίδα 1. Η σελίδα 1 δεν υπάρχει στη μνήμη, άρα έχουμε σφάλμα σελίδας και πρέπει να ελέγξουμε ποιά από τις σελίδες 2, 4 και 6 θα αργήσουμε περισσότερο να συναντήσουμε ξανά στο μέλλον.

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης OPT;

	2	4	6	1	4	5	6	2	1	6
	2	2	2							
		4	4							
			6							
ΣΦΑΛΜΑ?	X	X	X							

Παρατηρούμε, πως η σελίδα που θα συναντήσουμε ξανά το αργότερο στο μέλλον είναι η σελίδα 2. Άρα, θα την αφαιρέσουμε από τη μνήμη και στη θέση της θα βάλουμε την 1.

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης OPT;

2 4 6 1 4 5 6 2 1 6

2	2	2	<u>1</u>						
	4	4	4						
		6	6						
ΣΦΑΛΜΑ?	X	X	X						

Κατόπιν, συναντάμε τη σελίδα 4. Αυτή υπάρχει ήδη μέσα στη μνήμη, άρα δεν έχουμε σφάλμα σελίδας και δε χρειάζεται να αλλάξουμε τίποτα.

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης OPT;

2 4 6 1 4 5 6 2 1 6

2	2	2	1	1					
	4	4	4	4					
		6	6	6					
ΣΦΑΛΜΑ?	X	X	X	X					

Συνεχίζοντας, δεν έχουμε τη σελίδα 5 στη μνήμη. Άρα, **έχουμε σφάλμα σελίδας** και χρειάζεται να ελέγξουμε ποιά από τις σελίδες 1, 4 και 6 θα αργήσουμε περισσότερο να συναντήσουμε ξανά στο μέλλον.

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης OPT;

	2	4	6	1	4	5	6	2	1	6
	2	2	2	1	1					
		4	4	4	4					
			6	6	6					
ΣΦΑΛΜΑ?	X	X	X	X						

Διακρίνουμε ότι **δεν συναντάμε ποτέ ξανά τη σελίδα 4**. Συνεπώς, σύμφωνα με τα βήματα του αλγορίθμου, χρειάζεται να την αφαιρέσουμε και στη θέση της να μπει η σελίδα 5.

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης OPT;

2 4 6 1 4 5 6 2 1 6

2	2	2	1	1	1				
	4	4	4	4	<u>5</u>				
		6	6	6	6				
ΣΦΑΛΜΑ?	X	X	X		X				

Προχωρώντας, βλέπουμε πως η σελίδα 6 υπάρχει στη μνήμη. Δεν υπάρχει σφάλμα σελίδας.

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης OPT;

2 4 6 1 4 5 6 2 1 6

2	2	2	1	1	1	1			
	4	4	4	4	5	5			
		6	6	6	6	6			
ΣΦΑΛΜΑ?	X	X	X		X				

Στη συνέχεια, η σελίδα 2 δεν υπάρχει στη μνήμη! Επομένως, **έχουμε σφάλμα σελίδας** και χρειάζεται να ελέγξουμε ποιά από τις σελίδες 1, 5 και 6 θα αργήσουμε περισσότερο να συναντήσουμε ξανά στο μέλλον.

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης OPT;

	2	4	6	1	4	5	6	2	1	6
	2	2	2	1	1	1	1			
		4	4	4	4	5	5			
			6	6	6	6	6			
ΣΦΑΛΜΑ?	X	X	X	X		X				

Από ότι φαίνεται, δεν θα συναντήσουμε ποτέ ξανά τη σελίδα 5. Άρα, θα την αντικαταστήσουμε με τη σελίδα 2.

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης OPT;

2 4 6 1 4 5 6 2 1 6

2	2	2	1	1	1	1	1		
	4	4	4	4	5	5	<u>2</u>		
		6	6	6	6	6	6		
ΣΦΑΛΜΑ?	X	X	X		X		X		

Έπειτα, συναντάμε τη σελίδα 1. Αυτή, υπάρχει ήδη στη μνήμη, οπότε δεν έχουμε σφάλμα σελίδας.

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης OPT;

2 4 6 1 4 5 6 2 1 6

2	2	2	1	1	1	1	1	1	
	4	4	4	4	5	5	2	2	
		6	6	6	6	6	6	6	
ΣΦΑΛΜΑ?	X	X	X		X		X		

Τέλος, συναντάμε τη σελίδα 6, η οποία επίσης υπάρχει μέσα στη μνήμη. Επομένως και πάλι δεν έχουμε σφάλμα σελίδας.

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων, πόσα σφάλματα σελίδας θα γίνουν αν τρέξετε τον αλγόριθμο αντικατάστασης OPT;

	2	4	6	1	4	5	6	2	1	6
	2	2	2	1	1	1	1	1	1	1
		4	4	4	4	5	5	2	2	2
			6	6	6	6	6	6	6	6
ΣΦΑΛΜΑ?	X	X	X	X		X		X		

Συνολικά, είχαμε 6 σφάλματα σελίδας!

Άσκηση εξάσκησης με τον αλγόριθμο OPT!

Για την ακολουθία αναφορών $\{7,1,2,5,4,7,8,5,2,4,8,3,2,7\}$, υπολογίστε πόσα σφάλματα μνήμης θα έχουμε αν χρησιμοποιήσουμε τον αλγόριθμο αντικατάστασης OPT με 4 πλαίσια.

ΛΥΣΗ

Για την ακολουθία αναφορών {7,1,2,5,4,7,8,5,2,4,8,3,2,7}, υπολογίστε πόσα σφάλματα μνήμης θα έχουμε αν χρησιμοποιήσουμε τον αλγόριθμο αντικατάστασης OPT με 4 πλαίσια.

	7	1	2	5	4	7	8	5	2	4	8	3	2	7
	7	7	7	7	7	7	8	8	8	8	8	3	3	7
		1	1	1	4	4	4	4	4	4	4	4	4	4
			2	2	2	2	2	2	2	2	2	2	2	2
				5	5	5	5	5	5	5	5	5	5	5
ΣΦΑΛΜΑ?	X	X	X	X	X		X					X		X

Συνολικά, είχαμε 8 σφάλματα σελίδας!

Σύνοψη

- **FIFO**: Από τις σελίδες μέσα στη μνήμη, αντικαθιστούμε εκείνη που μπήκε παλαιότερα.
- **LRU**: Από τις σελίδες μέσα στη μνήμη, αντικαθιστούμε εκείνη που συναντήσαμε τελευταία φορά πιο παλιά από όλες.
- **OPT**: Από τις σελίδες μέσα στη μνήμη, αντικαθιστούμε εκείνη που δεν πρόκειται να συναντήσουμε ξανά για το μεγαλύτερο διάστημα, ή εκείνη που δεν θα συναντήσουμε ποτέ ξανά (αν υπάρχει τέτοια περίπτωση).

ΤΕΛΟΣ

ΕΥΧΑΡΙΣΤΩ ΠΟΛΥ!