



Ουρές & Στοίβες

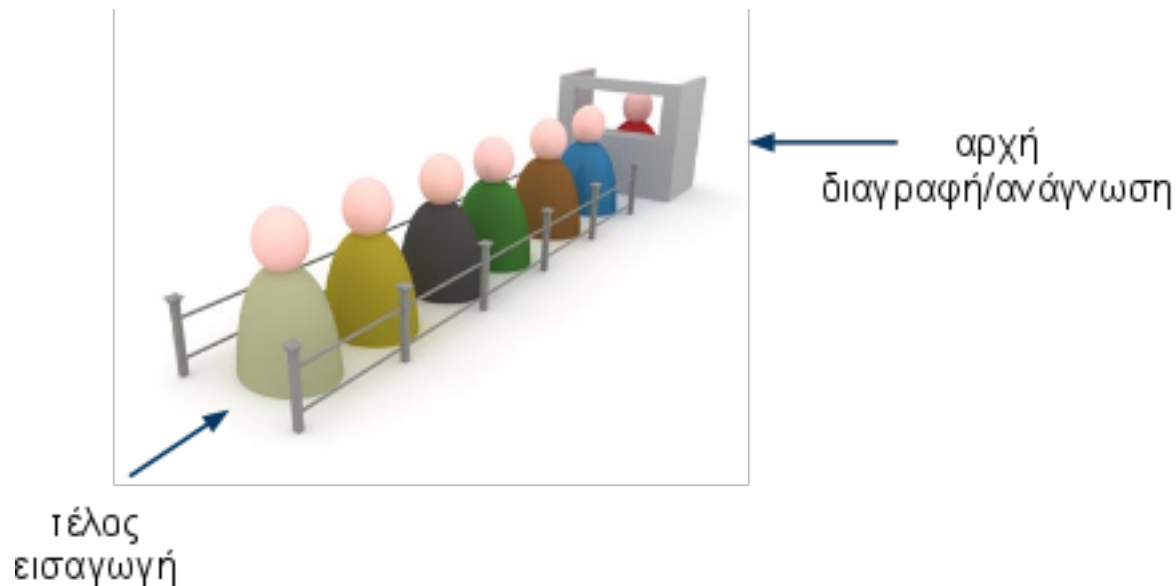
Διονύσης "dionyziz" Ζήνδρος

Camp Προετοιμασίας ΠΔΠ

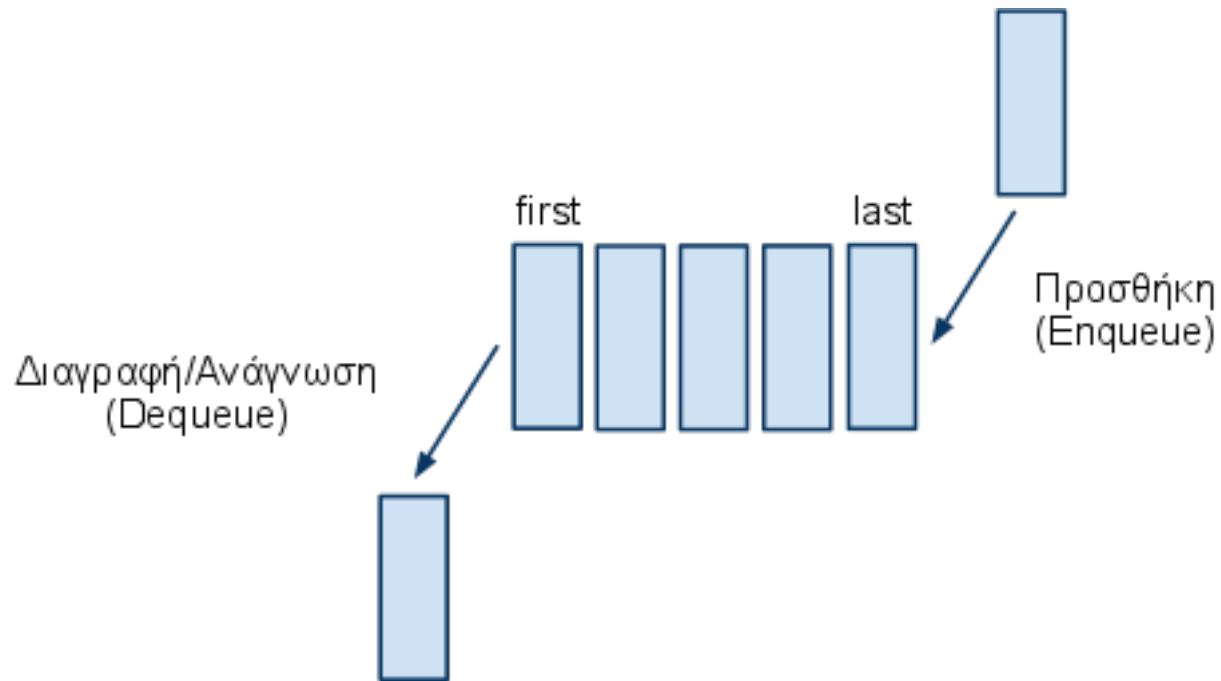
Άνοιξη 2011

Ουρές

- **Λίστες με περιορισμούς**
- Προσθέτω στοιχεία μόνο στο **τέλος**: Enqueue
- Διαγράφω και διαβάζω μόνο από την **αρχή**: Dequeue



Λειτουργίες ουράς



FIFO: First-in First-out



Γιατί;

- Δεν προσφέρουν κάτι παραπάνω από τις λίστες
- Προσφέρουν **εναλλακτικό τρόπο σκέψης**
- Σε πολλά προβλήματα δεν χρειαζόμαστε την πλήρη λειτουργία των λιστών
- Με ουρές γράφουμε **απλούστερα προγράμματα**
- Χρησιμοποιείται για να φτιάξουμε πιο περίπλοκες δομές

π.χ.

- Πελάτες που πρέπει να εξυπηρετηθούν σε τηλεφωνικό κέντρο.

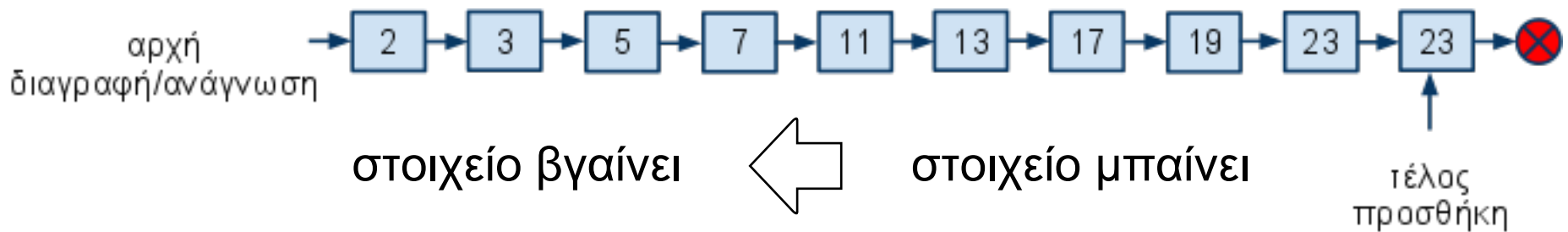


Υλοποίηση

- Υλοποιείται εύκολα χρησιμοποιώντας **λίστες**
- Στην πράξη χρησιμοποιούμε τις **έτοιμες ουρές** της C++
Μπορούμε να τις υλοποιήσουμε και με **πίνακα**

Υλοποίηση με λίστα

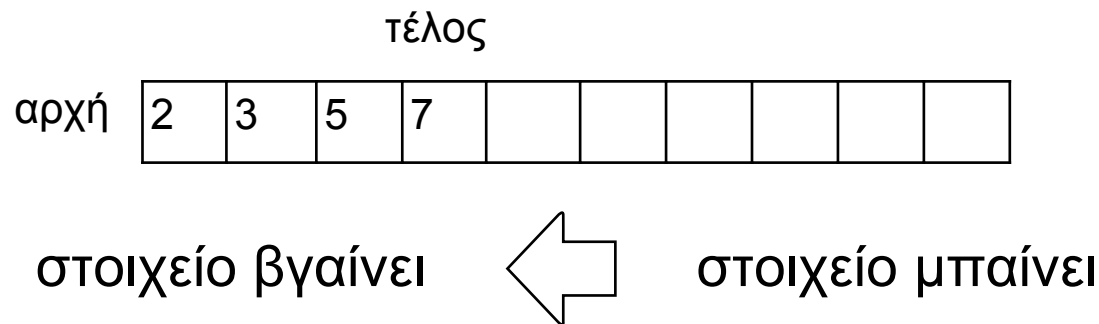
- Αποθηκεύουμε head και tail
- Προσθέτουμε στο τέλος σε $O(1)$
- Διαγράφουμε ή διαβάζουμε από την αρχή σε $O(1)$



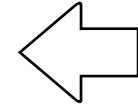
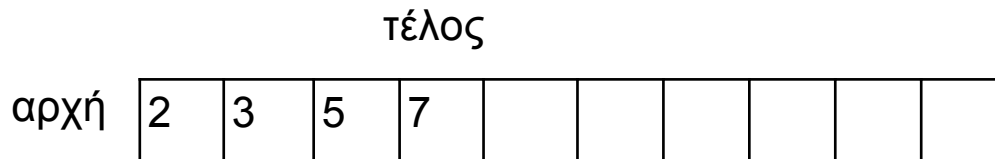
- Τα στοιχεία κινούνται από **το τέλος** προς **την αρχή**
- Η φορά της λίστας είναι **αντίθετη** από τη φορά επεξεργασίας

Υλοποίηση με πίνακα

- Αποθηκεύουμε τα στοιχεία σε πίνακα
- Αποθηκεύουμε το index του front και του back
- Δεν μετακινούμε τα στοιχεία

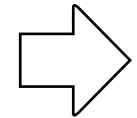
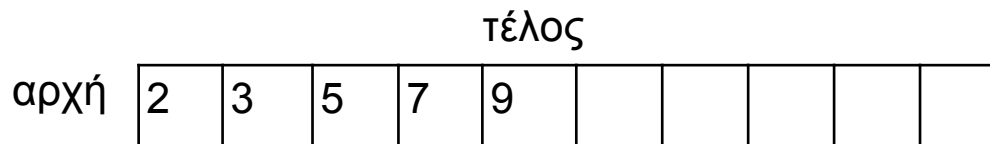


Προσθήκη & Διαγραφή



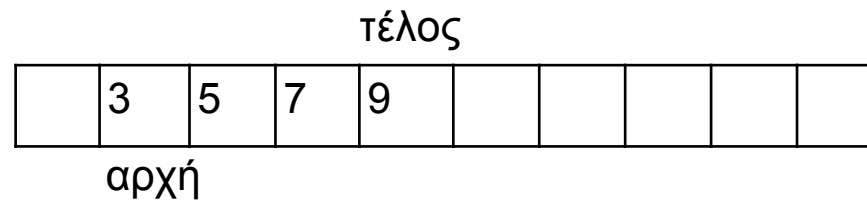
φορά επεξεργασίας

- Προσθήκη: **Enqueue**(9)



φορά μετακίνησης

- Διαγραφή/Ανάγνωση: **Dequeue**()



Στοίβες

- Πάλι **λίστες** με **περιορισμούς**
- Προσθέτω στοιχεία μόνο στην αρχή (Push)
- Διαγράφω και διαβάζω μόνο από την **αρχή** (Pop)

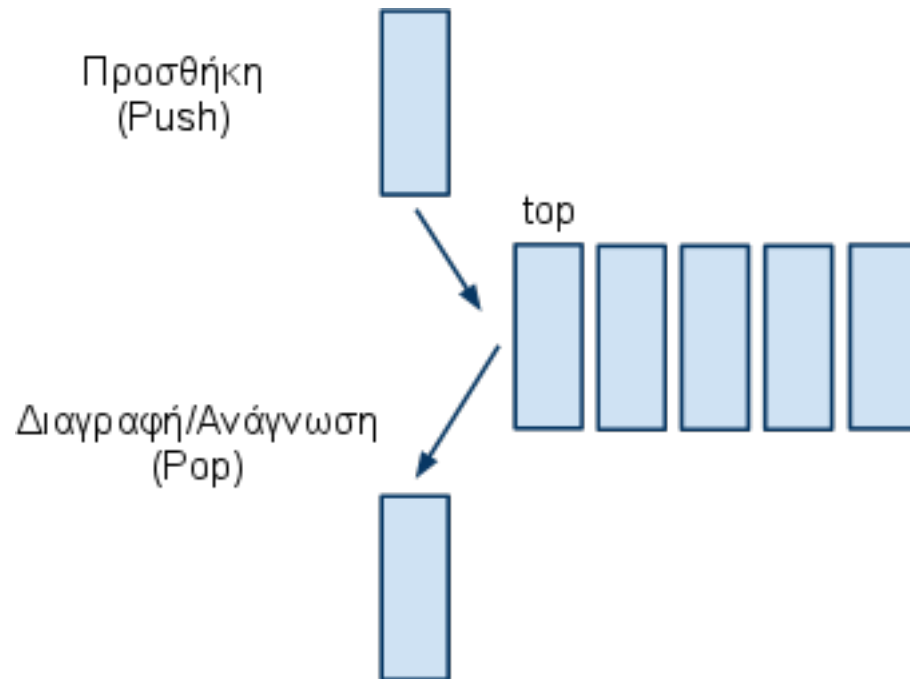




Γιατί;

- Δεν προσφέρουν περισσότερα πράγματα από τις λίστες
- Βοηθάνε στον **τρόπο σκέψης**
- Χρήσιμες για μετατροπή **από αναδρομή σε επανάληψη**
- Χρήσιμη για την δυνατότητα αναδρομής γενικότερα
 - Ο compiler την χρησιμοποιεί για να αποθηκεύσει τις ενδιάμεσες κλήσεις
 - Αυτός είναι ο λόγος που **η αναδρομή είναι πιο αργή από την επανάληψη**

Λειτουργίες Στοίβας



LIFO: Last-in First-out

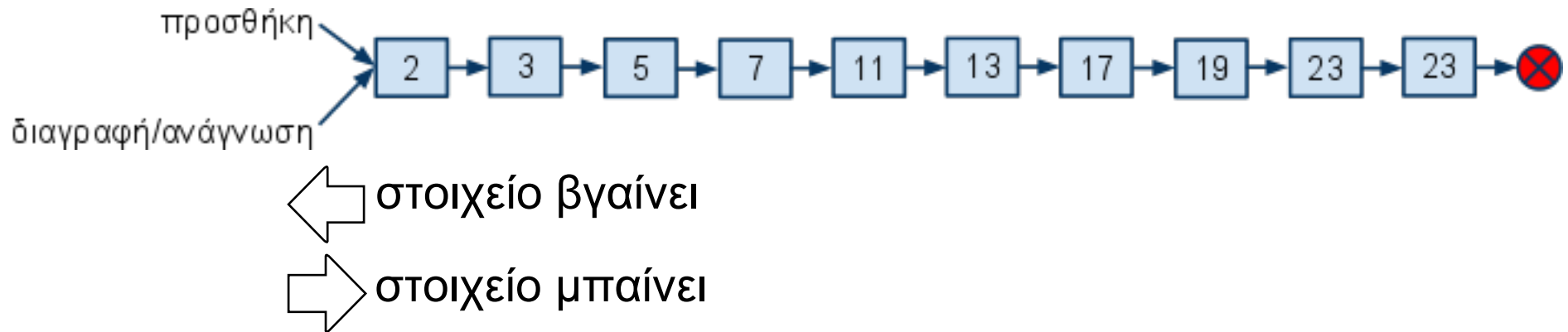


Υλοποίηση

- Υλοποιείται εύκολα χρησιμοποιώντας **λίστες**
- Στην πράξη χρησιμοποιούμε τις **έτοιμες στοίβες** της C++
Μπορούμε να τις υλοποιήσουμε και με **πίνακα**

Υλοποίηση με λίστα

- Αποθηκεύουμε head
- Προσθέτουμε στην αρχή σε $O(1)$
- Διαγράφουμε ή διαβάζουμε από την αρχή σε $O(1)$





Υλοποίηση με πίνακα

- Αποθηκεύουμε τα στοιχεία στον πίνακα
- Εισαγωγή στα δεξιά

2	3	5	7						
---	---	---	---	--	--	--	--	--	--

αρχή

- Προσθήκη: **Push**(9)

2	3	5	7	9					
---	---	---	---	---	--	--	--	--	--

αρχή

- Διαγραφή/Ανάγνωση: **Pop**()

2	3	5	7						
---	---	---	---	--	--	--	--	--	--

αρχή