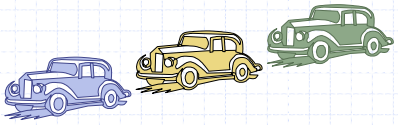
Ουρές



Κύρια σημεία για μελέτη

- Ο ΑΤΔ της Ουράς (§2.1.2)
- Υλοποίηση με ένα κυκλικό πινάκα (§2.1.2)
- Επεκτάσιμη ουρά βασισμένη σε πίνακα
- Το Interface της Ουράς σε Java

Ο ΑΤΔ της Ουράς

- Ο ΑΤΔ της Ουράς αποθηκεύει αυθαίρετα αντικείμενα
- Οι Εισαγωγές και οι Διαγραφές ακολουθούν το σχήμα first-in first-out
- Οι εισαγωγές γίνονται στο τέλος της ουράς και οι διαγραφές από την αρχή της ουράς
- Κύριες λειτουργίες της ουράς:
  - enqueue(αντικείμενο): εισάγει ένα στοιχείο στο τέλος της ουράς
  - αντικείμενο dequeue(): διαγράφει και επιστρέφει το στοιχείο στην αρχή της ουράς
- Βοηθητικές εργασίες της ουράς:
  - αντικείμενο front(): επιστρέφει το πρώτο στοιχείο χωρίς να το διαγράψει
  - integer size(): επιστρέφει τον αριθμό των στοιχείων που είναι αποθηκευμένα
  - boolean isEmpty(): δείχνει αν υπάρχουν ή όχι αποθηκευμένα στοιχεία
- Εξαιρέσεις
  - Η προσπάθεια εκτέλεσης dequeue ή front σε μια άδεια ουρά προκαλεί μια EmptyQueueException

Εφαρμογές των ουρών

- Άμεσες εφαρμογές
  - Λίστες αναμονής, γραφειοκρατία
  - Πρόσβαση σε μοιραζόμενους πόρους (π.χ, εκτυπωτής)
  - Πολυπρογραμματισμός
- Έμμεσες εφαρμογές
  - Βοηθητικές δομές δεδομένων για αλγόριθμους
  - Συστατικό άλλων δομών δεδομένων

Ουρά βασισμένη σε πίνακα

- Χρησιμοποίηση ενός πίνακα μεγέθους N με ένα κυκλικό τρόπο
- Δύο μεταβλητές κρατάνε στοιχεία για το πρώτο και τελευταίο στοιχείο της ουράς
  - f δείκτης του πρώτου στοιχείου
  - r δείκτης του αμέσως επόμενου μετά το τελευταίο στοιχείο
- Η θέση του r στον πίνακα παραμένει άδεια

Κανονική μορφή

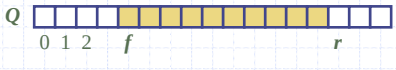
wrapped-around μορφή

Λειτουργίες της ουράς

- Χρησιμοποιούμε τον τελεστή modulo (υπόλοιπο της διαίρεσης)

Algorithm *size*()
  return $(N - f + r) \bmod N$

Algorithm *isEmpty*()
  return $(f = r)$



Λειτουργίες της ουράς (συνεχ.)

- Η λειτουργία enqueue προκαλεί μια exception όταν ο πίνακας είναι γεμάτος
- Αυτή η exception εξαρτάται από την υλοποίηση

Algorithm *enqueue*(o)
  if *size*() $= N - 1$ **then**
    throw *FullQueueException*
  else
    $Q[r] \leftarrow o$
    $r \leftarrow (r + 1) \bmod N$

Λειτουργίες της ουράς (συνεχ.)

- Η λειτουργία dequeue προκαλεί μια exception όταν ο πίνακας είναι άδειος
- Αυτή η exception είναι καθορισμένη στον ΑΤΔ της ουράς

Algorithm *dequeue*()
  if *isEmpty*() **then**
    throw *EmptyQueueException*
  else
    $o \leftarrow Q[f]$
    $f \leftarrow (f + 1) \bmod N$
    return o

Επεκτάσιμη ουρά βασισμένη σε πίνακα

- Σε μια λειτουργία enqueue, όταν ο πίνακας είναι γεμάτος, αντί να προκαλείται μια exception , μπορούμε να αντικαταστήσουμε τον πίνακα με ένα μεγαλύτερο
- Όμοια με ότι κάναμε για τη στοίβα βασισμένη σε πίνακα
- Η λειτουργία enqueue έχει ακαθόριστο χρόνο εκτέλεσης
  - $O(n)$ με τη στρατηγική προσαύξησης
  - $O(1)$ με τη στρατηγική διπλασιασμού

Το Interface της Ουράς σε Java

- Το interface της Java που αντιστοιχεί στον ΑΤΔ της ουράς που περιγράψαμε
- Απαιτεί τον καθορισμό της κλάσης EmptyQueueException
- Δεν υπάρχει αντίστοιχη ενσωματωμένη κλάση σε Java

```
public interface Queue {
  public int size();
  public boolean isEmpty();
  public Object front()
      throws EmptyQueueException;
  public void enqueue(Object o);
  public Object dequeue()
      throws EmptyQueueException;
}
```