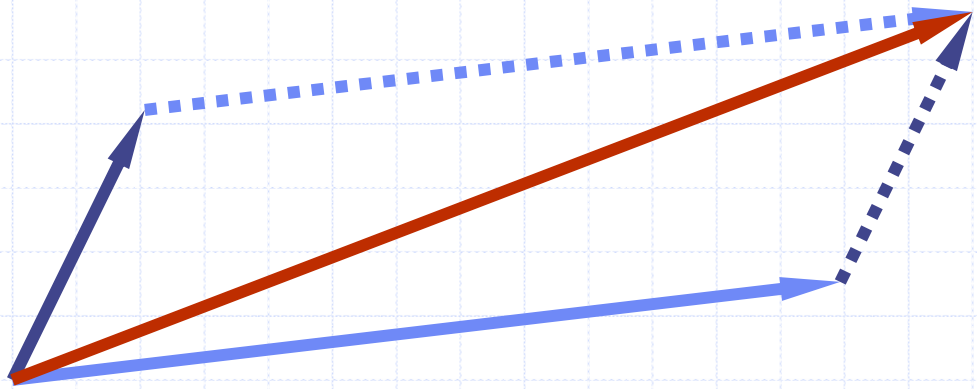
Διανύσματα



Κύρια σημεία για μελέτη

- Ο ΑΤΔ του διανύσματος (§2.2.1)
- Υλοποίηση βασισμένη σε πίνακα (§2.2.1)

Ο ΑΤΔ του διανύσματος

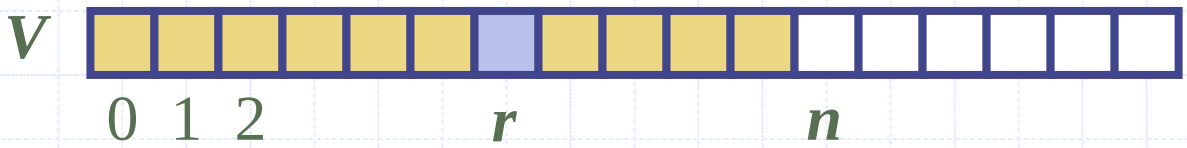
- Ο ΑΤΔ του διανύσματος αποτελεί επέκταση της ιδέας του πίνακα, αποθηκεύοντας μία ακολουθία από αυθαίρετα αντικείμενα
- Ένα στοιχείο μπορεί να προσπελαστεί,να εισαχθεί ή να διαγραφεί καθορίζοντας την τάξη του (αριθμός των προηγούμενων στοιχείων του)
- Προκαλείται μια exception αν καθοριστεί μία λάθος τάξη (π.χ. μια αρνητική τάξη)
- Κύριες λειτουργίες διανύσματος:
  - αντικείμενο elemAtRank(integer r): επιστρέφει ένα στοιχείο της τάξης r χωρίς να το διαγράψει
  - αντικείμενο replaceAtRank(integer r, αντικείμενο o): αντικαθιστά το στοιχείο της τάξης με το o και επιστρέφει το παλιό στοιχείο
  - insertAtRank(integer r, αντικείμενο o): εισάγει ένα νέο στοιχείο o με τάξη r
  - αντικείμενο removeAtRank(integer r): διαγράφει και επιστρέφει το στοιχείο με τάξη r
- Επιπλέον λειτουργίες size() και isEmpty()

Εφαρμογές των διανυσμάτων

- Άμεσες εφαρμογές
  - Ταξινομημένη συλλογή αντικειμένων (στοιχειώδης βάση δεδομένων)
- Έμμεσες εφαρμογές
  - Βοηθητικές δομές δεδομένων για αλγόριθμους
  - Συστατικό άλλων δομών δεδομένων

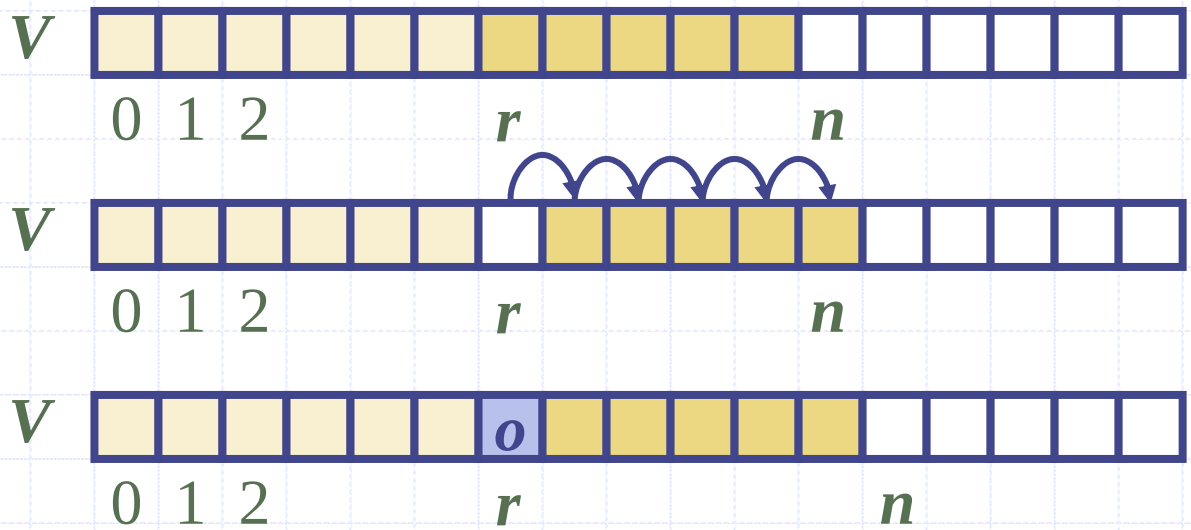
Διάνυσμα βασισμένο σε πίνακα

- Χρήση ενός πίνακα V μεγέθους N
- Μία μεταβλητή n ανιχνεύει το μέγεθος του διανύσματος (αριθμός των αποθηκευμένων στοιχείων)
- Η λειτουργία $elemAtRank(r)$ εκτελείται σε χρόνο $O(1)$ επιστρέφοντας $V[r]$



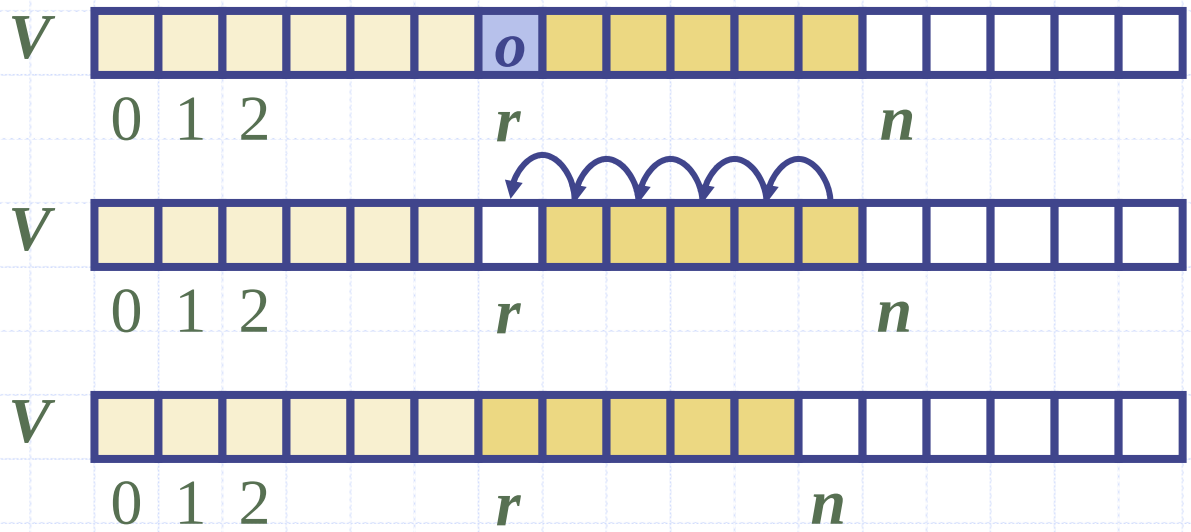
Εισαγωγή

- Στη λειτουργία ***insertAtRank***(r, o), για να φτιάξουμε χώρο για ένα νέο στοιχείο χρειάζεται να κινήσουμε προς τα μπροστά $n - r$ στοιχεία $V[r], \ldots, V[n-1]$
- Στη χειρότερη περίπτωση ($r = 0$), αυτό παίρνει χρόνο $O(n)$



Διαγραφή

- Στη λειτουργία ***removeAtRank***(r), για να γεμίσουμε το κενό που άφησε το διαγραμμένο στοιχείο χρειάζεται να κινήσουμε προς τα πίσω $n - r - 1$ στοιχεία $V[r + 1], \ldots, V[n - 1]$
- Στη χειρότερη περίπτωση ($r = 0$), αυτό παίρνει χρόνο $O(n)$



Απόδοση

- Στην υλοποίηση ενός διανύσματος βασισμένη σε πίνακα
  - Ο χώρος που χρησιμοποιείται από τη δομή δεδομένων είναι $O(n)$
  - ***size***, ***isEmpty***, ***elemAtRank*** και ***replaceAtRank*** τρέχουν σε χρόνο $O(1)$
  - ***insertAtRank*** και ***removeAtRank*** τρέχουν σε χρόνο $O(n)$
- Αν χρησιμοποιούμε τον πίνακα με κυκλικό τρόπο, ***insertAtRank***(0) και ***removeAtRank***(0) τρέχουν σε χρόνο $O(1)$
- Σε μια λειτουργία ***insertAtRank***, όταν ο πίνακας είναι γεμάτος, αντί να προκαλείται μία exception, μπορούμε να αντικαταστήσουμε τον πίνακα με έναν μεγαλύτερο