Ουρές Προτεραιότητας

Sell	100	IBM	$122
Sell	300	IBM	$120
Buy	500	IBM	$119
Buy	400	IBM	$118

Περιγραφή και Υλικό Ανάγνωσης

- Ο Αφηρημένος Τύπος Δεδομένων της Ουράς Προτεραιότητας (PriorityQueue ADT) (§2.4.1)
- Σχέση απόλυτης ταξινόμησης (§2.4.1)
- Ο Αφηρημένος Τύπος Δεδομένων του Συγκριτή (Comparator ADT) (§2.4.1)
- Ταξινόμηση με ουρά προτεραιότητας (§2.4.2)
- Selection-sort (§2.4.2)
- Insertion-sort (§2.4.2)

Ο ΑΤΔ της Ουράς Προτεραιότητας

- Μια ουρά προτεραιότητας αποθηκεύει μια συλλογή αντικειμένων
- Ένα αντικείμενο είναι ένα ζευγάρι (κλειδί, στοιχείο)
- Κύριες πράξεις της ουράς προτεραιότητας
  - insertItem(k, o)
    εισάγει ένα αντικείμενο με κλειδί k και στοιχείο o
  - removeMin()
    απομακρύνει το αντικείμενο με το μικρότερο κλειδί και επιστρέφει το στοιχείο
- Βοηθητικές πράξεις
  - minKey(k, o)
    επιστρέφει, αλλά δεν απομακρύνει, το μικρότερο κλειδί ενός αντικειμένου
  - minElement()
    επιστρέφει, αλλά δεν απομακρύνει, το στοιχείο του αντικειμένου με το μικρότερο κλειδί
  - size(), isEmpty()
- Εφαρμογές:
  - Standby flyers
  - Πλειστηριασμοί
  - Χρηματιστήριο

Σχέση Απόλυτης Ταξινόμησης

- Κλειδιά σε μια ουρά προτεραιότητας μπορούν να είναι αυθαίρετα αντικείμενα πάνω στα οποία ορίζεται μια σειρά
- Δύο ξεχωριστά αντικείμενα σε μια ουρά προτεραιότητας μπορούν να έχουν το ίδιο κλειδί
- Μαθηματική έννοια της σχέσης απόλυτης ταξινόμησης $\leq$
  - Ανακλαστική ιδιότητα:
    $x \leq x$
  - Αντισυμμετρική ιδιότητα:
    $x \leq y \wedge y \leq x \Rightarrow x = y$
  - Μεταβατική ιδιότητα:
    $x \leq y \wedge y \leq z \Rightarrow x \leq z$

Ο ΑΤΔ του Συγκριτή

- Ένας συγκριτής περικλείει την πράξη της σύγκρισης δύο αντικειμένων σύμφωνα με μια δεδομένη σχέση απόλυτης ταξινόμησης
- Μια γενική ουρά προτεραιότητας χρησιμοποιεί ένα βοηθητικό συγκριτή
- Ο συγκριτής είναι ανεξάρτητος από τα κλειδιά που συγκρίνονται
- Όταν η ουρά προτεραιότητας χρειάζεται να συγκρίνει δύο κλειδιά, χρησιμοποιεί τον συγκριτή
- Πράξεις του ΑΤΔ του συγκριτή, όλες με επιστρεφόμενο τύπο Boolean
  - isLessThan(x, y)
  - isLessThanOrEqualTo(x,y)
  - isEqualTo(x,y)
  - isGreaterThan(x, y)
  - isGreaterThanOrEqualTo(x,y)
  - isComparable(x)

Ταξινόμηση με Ουρά Προτεραιότητας (PQ-sort)

- Μπορούμε να χρησιμοποιήσουμε μια ουρά προτεραιότητας για να ταξινομήσουμε ένα σύνολο από συγκρίσιμα στοιχεία
  1. Εισήγαγε τα στοιχεία ένα προς ένα με μια σειρά από πράξεις insertItem(e, e)
  2. Απομάκρυνε τα στοιχεία σε ταξινομημένη σειρά με μια σειρά από πράξεις removeMin()
- Ο χρόνος εκτέλεσης αυτής της μεθόδου ταξινόμησης εξαρτάται από την υλοποίηση της ουράς προτεραιότητας

Algorithm *PQ-Sort*(*S*, *C*)
Input sequence *S*, comparator *C* for the elements of *S*
Output sequence *S* sorted in increasing order according to *C*
P ← priority queue with comparator *C*
while ¬*S.isEmpty* ()
e ← *S.remove* (*S. first* ())
P.insertItem(*e*, *e*)
while ¬*P.isEmpty*()
e ← *P.removeMin*()
S.insertLast(*e*)

Sequence-based Priority Queue

- Υλοποίηση με μια μη-ταξινομημένη ακολουθία
  - Αποθήκευσε τα αντικείμενα της ουράς προτεραιότητας σε μια ακολουθία βασισμένη σε λίστα, σε αυθαίρετη σειρά
- Απόδοση:
  - Η insertItem χρειάζεται $O(1)$ χρόνο καθώς μπορούμε να εισάγουμε κάθε αντικείμενο στην αρχή ή στο τέλος της ακολουθίας
  - Οι removeMin, minKey και minElement χρειάζονται $O(n)$ χρόνο καθώς πρέπει να διασχίσουμε όλη την ακολουθία για να βρούμε το μικρότερο κλειδί
- Υλοποίηση με μια ταξινομημένη ακολουθία
  - Αποθήκευσε τα αντικείμενα της ουράς προτεραιότητας σε μια ακολουθία, ταξινομημένα με βάση το κλειδί τους
- Απόδοση:
  - Η insertItem χρειάζεται $O(n)$ χρόνο καθώς πρέπει να βρούμε τη θέση που θα εισάγουμε το αντικείμενο
  - Οι removeMin, minKey και minElement χρειάζονται $O(1)$ χρόνο καθώς το μικρότερο κλειδί βρίσκεται πάντα στην αρχή της ακολουθίας

Selection-Sort (ταξινόμηση διαλογής)

- Η selection-sort είναι μια παραλλαγή της PQ-sort όπου η ουρά προτεραιότητας έχει υλοποιηθεί με μια μη-ταξινομημένη ακολουθία
- Χρόνος εκτέλεσης της Selection-sort:
  1. Η εισαγωγή των στοιχείων στην ουρά προτεραιότητας με n πράξεις insertItem χρειάζεται $O(n)$ χρόνο
  2. Η απομάκρυνση των στοιχείων σε ταξινομημένη σειρά από την ουρά προτεραιότητας με n πράξεις removeMin χρειάζεται χρόνο ανάλογο του

$$1 + 2 + \ldots + n$$

- Η Selection-sort εκτελείται σε $O(n^2)$ χρόνο

Insertion-Sort (ταξινόμηση εισαγωγής)

- Η Insertion-sort είναι μια παραλλαγή της PQ-sort όπου η ουρά προτεραιότητας έχει υλοποιηθεί με μια ταξινομημένη ακολουθία
- Χρόνος εκτέλεσης της Insertion-sort:
  1. Η εισαγωγή των στοιχείων στην ουρά προτεραιότητας με n πράξεις insertItem χρειάζεται χρόνο ανάλογο του
     $$1 + 2 + \ldots + n$$
  2. Η απομάκρυνση των στοιχείων σε ταξινομημένη σειρά από την ουρά προτεραιότητας με n πράξεις removeMin χρειάζεται $O(n)$ χρόνο
- Η Insertion-sort εκτελείται σε $O(n^2)$ χρόνο

In-place Insertion-sort

- Αντί να χρησιμοποιήσουμε μια εξωτερική δομή δεδομένων, μπορούμε να υλοποιήσουμε selection-sort και insertion-sort in-place (δηλαδή πάνω στην ίδια δομή)
- Ένα τμήμα της εισαγόμενης ακολουθίας εξυπηρετεί ως η ουρά προτεραιότητας
- Για in-place insertion-sort
  - Κρατάμε ταξινομημένο το αρχικό τμήμα της ακολουθίας
  - Μπορούμε να χρησιμοποιήσουμε τη swapElements αντί να τροποποιήσουμε την ακολουθία