

ΕΡΓΑΣΙΑ 2

Δημήτρης Πεντάρης 3220159

Κωνσταντίνος Καρτάλος 3220076

Μέρος Α:

Χρησιμοποιήσαμε τον αλγόριθμο HeapSort για την υλοποίηση του πρώτου μέρους. Βασική ιδέα της HeapSort είναι να αξιοποιήσει τις ιδιότητες ενός Heap (σωρός). Αρχικά παίρνουμε τα δεδομένα από το αρχείο και δημιουργούμε ένα Heap του οποίου οι κόμβοι περιέχουν τις πόλεις. Εφόσον το πρώτο σε προτεραιότητα στοιχείο είναι η ρίζα του Heap αρκεί κάθε φορά να αφαιρούμε αυτό και αμέσως μετά να αποκαταστήσουμε την ιδιότητα της σωρού στην δομή μας. Για να γίνει αυτό αρκεί να συγκρίνουμε το κάθε στοιχείο με τον πατέρα του και τα παιδιά του και ανάλογα να κάνουμε τις αντιμεταθέσεις μέχρι το στοιχείο να μην είναι μικρότερο από τον πατέρα και μεγαλύτερο από τα παιδιά του (σε προτεραιότητα), μιας και μιλάμε για δυαδικό δέντρο. Το τελικό αποτέλεσμα είναι τα δεδομένα ταξινομημένα από το μικρότερο στο μεγαλύτερο.

Μέρος Β:

Για την υλοποίηση της remove χρειαζόμαστε μια βοηθητική μεταβλητή στην οποία θα αποθηκεύουμε το στοιχείο που θέλουμε να διαγράψουμε. Αμέσως μετά ανταλλάσσουμε θέσεις του στοιχείου που θέλουμε να διαγράψουμε με το τελευταίο σε προτεραιότητα και αμέσως μετά το διαγράφουμε. Για να αποκατασταθεί η ιδιότητα της σωρού χρησιμοποιούμε τις συναρτήσεις swim και sink, οι οποίες μεταφέρουν το στοιχείο στη σωστή του θέση.

Μέρος Γ:

Το Μέρος Γ αξιοποιεί αποτελεσματικά ό,τι υλοποιήσαμε στο Μέρος Β φτιάχνοντας μια γρηγορότερη και κομψότερη προσέγγιση του προγράμματος του μέρους Α. Το πρόγραμμα αποθηκεύει δυναμικά τα δεδομένα και ποτέ δεν ξεπερνάει το όριο των k αποθηκευμένων στοιχείων. Μόλις διαβάσει k στοιχεία ξεκινά να ελέγχει τα στοιχεία που διαβάζει ώστε πάντα να αποθηκεύει τα k με τη μεγαλύτερη προτεραιότητα από αυτά που έχει διαβάσει μέχρι στιγμής. Αυτό

επιτυγχάνεται με τη χρήση μιας μεταβλητής στην οποία αποθηκεύεται το στοιχείο με τη μικρότερη προτεραιότητα από τα k αποθηκευμένα, καθώς όποιο στοιχείο διαβάζεται μετέπειτα, συγκρίνεται με αυτό. Τέλος, τυπώνει τα στοιχεία με την επιθυμητή σειρά. Οι πολυπλοκότητες των μεθόδων είναι ίδιες με τις επιθυμητές της εκφώνησης. Η σωστή σειρά των δεδομένων είναι αποτέλεσμα του τρόπου κατασκευής των μεθόδων insert και remove. Ο αλγόριθμος του μέρους Γ είναι γρηγορότερος του αλγορίθμου του μέρους Α στην περίπτωση που το k είναι πολύ μικρότερο του συνολικού αριθμού πόλεων. Αυτό συμβαίνει διότι δεν αποθηκεύει περιττά στοιχεία και άρα χρησιμοποιείται μικρότερη σωρός(λιγότερες συγκρίσεις).