

Η ουρά μηνυμάτων περιέχει μηνύματα του τύπου:

```
typedef struct {  
    int flag; // γίνεται 1 όταν το τμήμα του πίνακα είναι ταξινομημένο  
    int start; // η αρχή του πίνακα ή υποπίνακα  
    int tail; // το τέλος του πίνακα ή υποπίνακα  
    double *a; // δείκτης στο πίνακα  
}Info;
```

Η ουρά είναι δομή του τύπου:

```
typedef struct {  
    Info array[MESSAGES]; // ο πίνακας των μηνυμάτων  
    int qin, qout; // δείκτες στη θέση που θα μπει ή θα βγει το μήνυμα αντίστοιχα  
    int count; // πόσα μηνύματα έχει η ουρά  
}CBUFFER;
```

Η κύρια συνάρτηση αρχικοποιεί την ουρά και βάζει μέσα το πρώτο μήνυμα. Κατόπιν δημιουργεί τα νήματα και μετά ελέγχει στην ουρά τα μηνύματα ώστε να πάρει αυτά που αφορούν ταξινομημένα τμήματα του πίνακα και μετράει πόσα από τα στοιχεία του πίνακα έχουν ταξινομηθεί. Όταν ταξινομηθούν όλα τότε βάζει στην ουρά μηνύματα τερματισμού για τα νήματα και μετά περιμένει τα νήματα να τελειώσουν.

Τέλος ελέγχει το πίνακα για σωστή ταξινόμηση και ελευθερώνει τους πόρους που έχουν δεσμευτεί.

Το κάθε νήμα παίρνει ένα μήνυμα και:

- αν περιέχει ταξινομημένο πίνακα το ξαναβάζει στην ουρά
- αν είναι μήνυμα τερματισμού τερματίζει
- αν περιέχει μη ταξινομημένο πίνακα τότε τον χωρίζει στα δύο με κλήση της partition και βάζει 2 μηνύματα στην ουρά
- αν το μήνυμα έχει πίνακα με λιγότερα ή ίσα στοιχεία με την CUTOFF τότε τον ταξινομεί με την insort και βάζει ένα μήνυμα στην ουρά με ένδειξη ότι έχει ταξινομηθεί

Από διάφορες εκτελέσεις είδα ότι το πρόγραμμα ταξινομεί τελικά τον πίνακα

Νικολάου Ευάγγελος AM Π2014150. Πηγες : ίντερνετ, σημειώσεις μαθήματος