Κατ’οίκον Εργασία 5 – Σκελετοί Λύσεων

Άσκηση 1

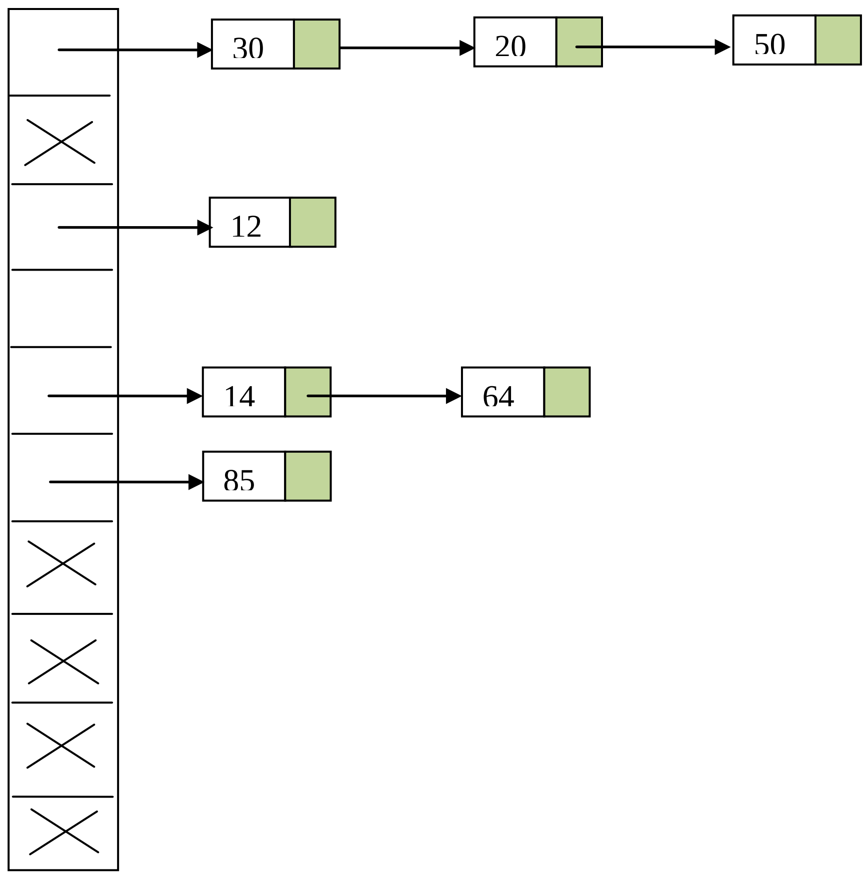
Εφαρμόζουμε παραλλαγή του αλγόριθμου του Dijkstra με τη διαφορά ότι ως απόσταση ανάμεσα σε δύο κορυφές θεωρούμε το μέγιστο βάρος ακμής και όχι το άθροισμα των βαρών όλων των ακμών.

```
Bottleneck(graph G, vertex v){
     heap Q;
     for (u = 0; u <= G->size – 1, u++)
          bd[u] = ∞;
     bd[v] = 0;
     S=∅;
     Q=V;
     while (!IsEmpty(Q))
          u=DeleteMin(Q);
          S=S∪{u};
          for (p = G->matrix[u]; p != NULL; p = p->next)
               w = p->vertex;
               if ( bd[w] > max( bd[u], weight(u,w) ) )
                    bd[w]= max( bd[u], weight(u,w) );
     }
```

Επανάληψη του πιο πάνω αλγόριθμου για κάθε κορυφή του γράφου θα μας δώσει την απόσταση μποτιλιαρίσματος ανάμεσα σε κάθε ζεύγος κορυφών του.

<u>Άσκηση 2</u>

(α)



(β)

50		**12**	**14**	**64**	**85**				

Η εισαγωγή των 20 και 30 δεν είναι εφικτή.

(γ)

50		**12**	**20**	**64**	**85**		**14**		**30**

(δ)

50	**14**	**12**	**20**	**64**	**85**				

Η εισαγωγή του 30 δεν είναι εφικτή.

Η λύση που αναφέρεται στον επανακατακερματισμό παραλείπεται.

Άσκηση 3

(α) Εκμεταλλευόμαστε τη βασική ιδέα του αλγόριθμου ταξινόμησης QuickSort ως εξής:

- Πάρε ένα κλειδί από τον θυρωρό και σύγκρινέ το με όλες τις κλειδαριές. Σημείωσε το σύνολο των κλειδαριών που είναι μεγαλύτερες από το κλειδί, έστω B, και το σύνολο των κλειδαριών που είναι μικρότερες, έστω S. Ξεκλείδωσε την αντίστοιχη πόρτα, έστω P.
- Επέστρεψε το κλειδί στον θυρωρό και διαδοχικά πάρε από αυτόν όλα τα υπόλοιπα κλειδιά και σύγκρινέ τα με την πόρτα P. Σημείωσε τα κλειδιά που είναι μεγαλύτερα από την κλειδαριά, έστω X και τα κλειδιά που είναι μικρότερα, έστω Y.
- Επανάλαβε τη διαδικασία στο σύνολο κλειδαριών B με τα κλειδιά X και στο σύνολο των κλειδαριών S με τα κλειδιά Y.

Παρατηρούμε πως ο χρόνος εκτέλεσης της διαδικασίας είναι όμοιο ς με αυτό ν του αλγορίθμου QuickSort (και οι δύο χρόνοι δίνονται από την ίδια αναδρομική εξίσωση).

Επομένως, και σε αυτή την περίπτωση, ο αλγόριθμος είναι της τάξης $O(n^2)$ στη χείριστη περίπτωση και $O(n \lg n)$ στη μέση περίπτωση.

(β) Παρατηρούμε ότι υπάρχουν n! διαφορετικοί τρόποι να ταιριάξουν n κλειδιά σε n κλειδαριές. Κάθε προσπάθεια για ταίριασμα έχει τρία δυνατά αποτελέσματα (το κλειδί να είναι μεγαλύτερο, μικρότερο ή ίσο με την κλειδαριά). Οποιοσδήποτε αλγόριθμος για επίλυση του προβλήματος μπορεί να περιγραφεί ως ένα δένδρο αποφάσεων με n! φύλλα, όπου ο χρόνος εκτέλεσης για εύρεση λύσης του προβλήματος αντιστοιχεί στο ύψος του δένδρου. Μπορούμε να δείξουμε ότι ένα τριαδικό δένδρο με n! φύλλα έχει ύψος το πολύ $\log_3 n! \in \Theta(n \lg n)$.
Συμπέρασμα: Το $\Theta(n \lg n)$ είναι κάτω φράγμα για το πρόβλημα, δηλαδή, δεν υπάρχει αλγόριθμος με χρόνο εκτέλεσης μικρότερο από $\Theta(n \lg n)$ για επίλυση του προβλήματος..