

## Αλγόριθμοι και Πολυπλοκότητα

### 3η σειρά γραπτών ασκήσεων

Αλέξανδρος Μαυρογιάννης  
03109677 7ο εξάμηνο

---

#### Άσκηση 1: Προβολή Ταινιών

Στην περιγραφή του αλγορίθμου χρησιμοποιούνται λίστες για ευκρίνεια, αλλά μπορεί πολύ εύκολα να υλοποιηθεί μόνο με arrays.

Έστω ότι οι προτιμήσεις του  $i$ -οστού συνδρομητή αποτελούν ένα ζεύγος τιμών  $pair(i)$  (όπου  $pair(i).l$  και  $pair(i).r$  οι ξεχωριστές προτιμήσεις του ζεύγους) και η έξοδος είναι μια λίστα  $list$  από τέτοια ζεύγη, όπου τα πρώτα στοιχεία όλων των ζευγών αντιστοιχούν στο σύνολο των ταινιών που θα προσφέρονται το Σάββατο (και συμβολίζονται με τη λίστα  $list.l$ ) και τα δεύτερα στοιχεία στο σύνολο των ταινιών που προσφέρονται τη Κυριακή (και συμβολίζονται με την  $list.r$ ). Κάθε στοιχείο  $pair(i).l$  ή  $pair(i).r$  έχει μια boolean τιμή  $pair(i).pointed$  που ξεκινάει ως false.

Έστω ακόμα με βάση αυτές τις δομές τις ακόλουθες διαδικασίες:

- $swap(pair)$  που ανταλλάσσει τις τιμές μεταξύ των στοιχείων ενός ζεύγους
- $can\_add(list, pair)$  που εξετάζει σειριακά την λίστα και επιστρέφει:
  - 1 αν  $pair.l \notin list.l$  ,  $pair.r \notin list.r$  όπου το  $pair$  μπορεί να προστεθεί στη  $list$  όπως είναι.
  - 2 αν  $pair.r \in list.r$  ,  $pair.l \notin list.l$  και πρέπει να προστεθεί μόνο το  $pair.l$  . Στην περίπτωση αυτή, το στοιχείο  $list(i).r$  που αντιστοιχεί στο  $pair.r$  σημαδεύεται ως  $list(i).r.pointed=true$ .
  - 3 αν  $pair.l \in list.l$  ,  $pair.r \notin list.r$  και πρέπει να προστεθεί μόνο το  $pair.r$ . Στην περίπτωση αυτή, το στοιχείο  $list(i).l$  που αντιστοιχεί στο  $pair.l$  σημαδεύεται ως  $list(i).l.pointed=true$ .
  - 4 αν  $pair.l \in list.l$  και  $pair.r \in list.r$  , όπου δεν χρειάζεται να προστεθεί τίποτα. Στην περίπτωση αυτή,  $list(i).r.pointed=true$  και  $list(i).l.pointed=true$  .
  - 0 αν το  $pair$  δεν μπορεί να προστεθεί στην  $list$ , επειδή και τα δύο του στοιχεία υπάρχουν είτε στην  $list.l$  είτε στην  $list.r$  .
- $add(list, pair, mode)$  που προσθέτει στη  $list$  το  $pair$  με τον τρόπο που περιγράφεται από το  $mode$ , σύμφωνα με τους κωδικούς που επιστρέφει η  $can\_add$ .

- Η `can_swap(pair)` που επιστρέφει true αν `pair.l.pointed=false` και `pair.r.pointed=false`

Έτσι, ο ψευδοκώδικας είναι:

```
list=empty
for i =1 to n{
    temp=can_add(list,pair(i))           // pros8iki tis neas timis an ginetai
    if (temp != 0){
        add(list,pair(i),temp)
    } else {
        if can_swap(pair(i)){           // an de ginetai,
            swap(pair(i))               // elegxos an ginetai swap i nea timi
            temp=can_add(list,pair(i))
            if temp!=0{                 // swap, kai pros8iki an ginetai
                add(list,pair(i),temp)
            } else{
                swap(pair(i))           //undo swap
            }
        } else{                        // swap osa idi yparxonta zevgaria ginetai
            for j = (i-1) to 0 {
                if can_swap(list(j)){
                    swap(list(j))
                    temp=can_add(list,pair(i))
                    if temp!=0{
                        add(list,pair(i),temp)
                        break
                    } else{
                        swap(list(j))
                    }
                }
            }
        }
        // an den ginetai, pros8iki tainias kai tin alli mera
        if pair(i).l ∈ list.l
            add(list,pair(i),3)
        else if pair(i).l ∈ list.l
            add(list,pair(i),3)
    }
}
```

Λόγω της ανάγκης για  $j$  επαναλήψεις σε κάθε  $i$  και έως  $(j+1)$  κλήσεις της `can_swap` σε κάθε  $j$ , η πολυπλοκότητα είναι  $O(n^3)$

## Άσκηση 2: Μέτρηση Συντομότερων Μονοπατιών

Έστω ότι  $s$  είναι η αφετηρία. Τότε,  $D(v)$  είναι η ελάχιστη απόσταση μιας κορυφής  $v$  από την  $s$  και  $N(v)$  είναι ο αριθμός των ελαχίστων μονοπατιών μέχρι αυτή. Έχουμε:

$$D(s)=0, N(s)=1$$

$$D(v) = \min_{w \in \text{neighbours}} (D(w) + 1) \text{ και}$$

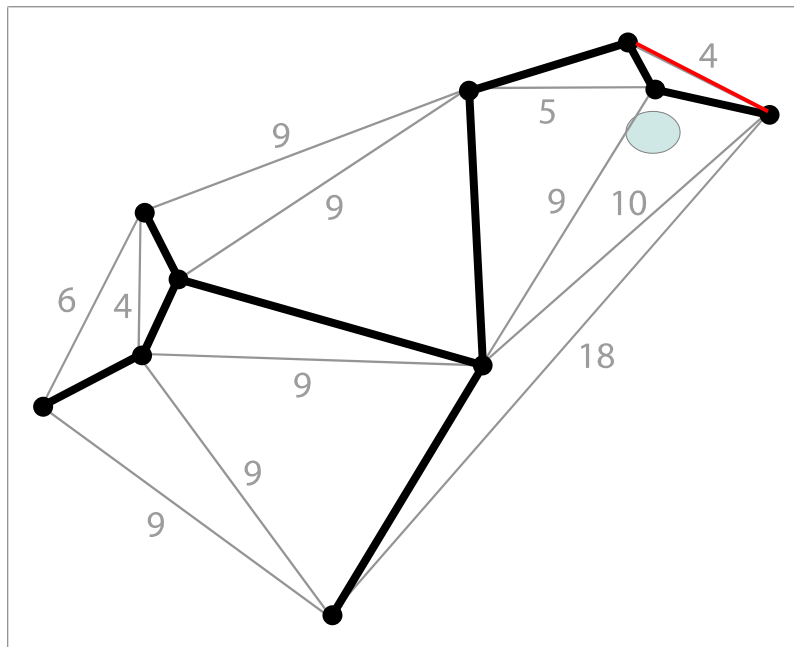
$$N(v) = \sum_{w \in \text{neighbours}: D(w)=D(v)} N(w)$$

όπου το σύνολο  $\text{neighbours}$  συμβολίζει το σύνολο των γειτονικών κορυφών με προορισμό προς τον κόμβο  $v$ .

Έτσι, υπολογίζουμε για κάθε κορυφή τις  $D$  και  $N$ , ξεκινώντας από την  $s$  και συνεχίζοντας με τις γειτονικές της, και έπειτα με τις γειτονικές τους, ώστε να καταλήψουμε στον προορισμό  $t$ , όπου το αποτέλεσμα θα είναι  $N(t)$ . Η χρονική πολυπλοκότητα είναι  $\Theta(V)$ , μιας και η κάθε κορυφή εξετάζεται μόνο μια φορά. Η χωρική πολυπλοκότητα είναι πάλι  $\Theta(V)$ , αφού χρειάζεται για κάθε κορυφή η αποθήκευση 2 τιμών.

## Άσκηση 3: Ελάχιστο Συνδετικό Δέντρο υπο περιορισμούς

- (α) Στον παρακάτω γράφο είναι τονισμένο το ελάχιστο συνδετικό δέντρο. Αν το σύνολο  $L$  συμπεριλαμβάνει όλα τα ήδη υπάρχοντα φύλλα αλλά και την κυκλωμένη κορυφή, τότε το ελάχιστο συνδετικό δέντρο υπο περιορισμούς θα πρέπει να συμπεριλαμβάνει την ακμή που είναι τονισμένη με κόκκινο, και άρα να είναι διαφορετικό από το ελάχιστο συνδετικό δέντρο του γράφου. Θεωρούμε πως α βάρη που δεν τονίζονται είναι όλα μικρότερα του 4.

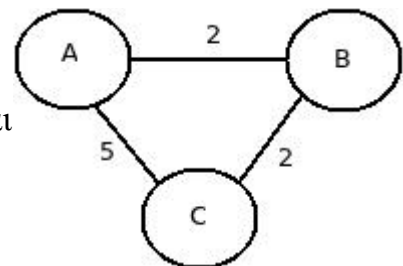


- (β) Αρκεί μια μικρή παραλλαγή στον αλγόριθμο του Kruskal:
- Έστω ένα δάσος  $F$  όπου κάθε κορυφή του αρχικού γράφου αποτελεί ένα δέντρο.
  - Έστω ακόμα το σύνολο  $S$  όλων των ακμών του γράφου.
  - Ταξινομούμε το  $S$  σε αύξουσα σειρά σε  $O(|E| \log |E|)$
  - Όσο το  $S$  δεν είναι κενό και όλα τα δέντρα του  $F$  δεν είναι συνδεδεμένα:
    - Αφαίρεση της ακμής με ελάχιστο βάρος από το  $S$
    - Αν η ακμή αυτή συνδέει δυο δέντρα στο  $F$ :
      - Αν κάποια κορυφή από τα άκρα της ακμής ανήκει στο  $L$  και αν αυτή παραμένει φύλλο μετά από την προσθήκη στο  $F$ , τότε προστίθεται στο  $F$ , αλλιώς αγνοείται
      - Αν δεν ανήκει στο  $L$ , προσθήκη της ακμής στο  $F$ .

Μόλις τερματίσει ο αλγόριθμος, θα έχει παραμείνει στο δάσος το ελάχιστο συνδετικό δέντρο όπου όλες οι κορυφές που ανήκουν στο  $L$  είναι φύλλα. Η πολυπλοκότητα είναι ίδια με τον αλγόριθμο του Kruskal και της ταξινόμησης,  $O(|E| \log |E|)$ .

#### Άσκηση 4: Μοναδικότητα Ελάχιστου Συνδετικού Δέντρου

- (α) Έστω ο γράφος του σχήματος με τρεις κορυφές  $A, B, C$  και τρεις ακμές με βάρη 2, 2 και 5. Το ΕΣΔ αποτελείται από τις ακμές με βάρη 2 και 2, και είναι μοναδικό αφού οποιοδήποτε άλλο συνδετικό δέντρο αποτελείται από ακμές με βάρη 2 και 5.



- (β) Υπάρχει ιδιότητα των ΕΣΔ που τονίζει πως αν για κάποια τομή  $(S, V \setminus S)$  του γράφου υπάρχει μοναδική ελάχιστη ακμή που διασχίζει την  $(S, V \setminus S)$ , τότε η ακμή αυτή ανήκει στο ΕΣΔ του γράφου αυτού. Έτσι, αν υπάρχουν μοναδικές ελάχιστες ακμές για όλες τις τομές του γράφου, τότε μόνο αυτές αποκλειστικά θα ανήκουν στο ΕΣΔ, και άρα το ΕΣΔ θα είναι μοναδικό αφού όλες οι ακμές που το αποτελούν είναι μοναδικές.

Για αντιπαράδειγμα, έστω πάλι ο απλός γράφος του (α) και  $S = \{B\}$ . Σε αυτή την περίπτωση, οι ακμές που διασχίζουν την  $(S, V \setminus S)$  είναι ίδιου βάρους, αλλά αποτελούν το ΕΣΔ του γράφου.

- (γ) Η συνθήκη είναι: Αν σε κάθε τομή του γράφου όλες οι ακμές ελάχιστου βάρους ανήκουν ταυτόχρονα στο ΕΣΔ, τότε το ΕΣΔ είναι μοναδικό.

Θεωρούμε ως  $A$  ένα ΕΣΔ το οποίο ικανοποιεί τη παραπάνω συνθήκη. Έστω ότι υπάρχει ένα άλλο ΕΣΔ,  $B$ , ίδιου βάρους με το  $A$  το οποίο διαφέρει από αυτό (τουλάχιστον) σε μια ακμή  $e_1$ . Αφού το  $A$  είναι ΕΣΔ που δεν περιλαμβάνει την  $e_1$ , θα υπάρχει σίγουρα στον γράφο ένας κύκλος  $C$ . Άρα, αφού το  $B$  είναι ΕΣΔ, θα πρέπει να υπάρχει μια ακμή  $e_2 \in C$  στο  $A$  που να μην ανήκει στο  $B$ .

Έτσι, παίρνουμε μια τομή  $(S, V \setminus S)$  την οποία διασχίζουν και οι δυο ακμές  $e_1, e_2$ . Αν  $e_1 < e_2$  τότε το  $B$  δεν είναι ΕΔΤ και αν  $e_2 < e_1$  τότε το  $A$  δεν είναι ΕΔΤ, λόγω του ερωτήματος (β).

Αν  $e_1 = e_2$  τότε δεν θα ανήκουν όλες οι μη-μοναδικές ελάχιστες ακμές στο ίδιο ΕΣΔ, άρα δεν θα ικανοποιείται η συνθήκη.

Επομένως, δεν γίνεται να υπάρχει άλλο ΕΣΔ, και άρα το  $A$  είναι μοναδικό.

(δ) Παραλλάσσοντας τον αλγόριθμο του Kruskal, έχουμε:

- Έστω ένα δάσος  $F$  όπου κάθε κορυφή του αρχικού γράφου αποτελεί ένα δέντρο.
- Έστω ακόμα το σύνολο  $S$  όλων των ακμών του γράφου.
- Ταξινομούμε το  $S$  σε αύξουσα σειρά σε  $O(|E| \log |E|)$
- Όσο το  $S$  δεν είναι κενό και όλα τα δέντρα του  $F$  δεν είναι συνδεδεμένα:
  - Αφαίρεση όλων των ακμών με το ελάχιστο βάρος από το  $S$  και τοποθέτηση σε ένα σύνολο  $T$
  - Αν όλες οι ακμές του  $T$  συνδέουν διαφορετικά δέντρα στο  $F$  ταυτόχρονα, προσθήκη της ακμής στο  $F$ .
  - Αν καμία ακμή του  $T$  δεν συνδέει δέντρα, τότε αυτή η ελάχιστη τιμή αγνοείται.
  - Αν μερικές ακμές συνδέουν δέντρα και μερικές όχι, τότε δεν υπάρχει μοναδικό ελάχιστο ΕΣΔ
- Αν βρέθηκε ΕΣΔ, τότε τότε αυτό είναι μοναδικό

Πάλι, η πολυπλοκότητα είναι ίδια με την ταξινόμηση και τον αλγόριθμο του Kruskal, δηλαδή  $O(|E| \log |E|)$ .

### Άσκηση 5: Υπολογισμός Ελάχιστου Συντακτικού Δέντρου με Διαγραφή Ακμών

- (α) Έστω  $A$  το ΕΣΔ του  $G$  που περιέχει την  $e$ , τότε εφόσον το  $A$  είναι ΕΣΔ, θα υπάρχει ακμή  $e'$  που ανήκει στον  $C$  και δεν ανήκει στο  $A$ . Αφού  $e > e'$ , το  $A$  δεν είναι ΕΣΔ, αφού υπάρχει ΕΣΔ με μικρότερο βάρος. Επομένως δεν γίνεται να υπάρχει ΕΣΔ που να περιέχει την  $e$ .
- (β) Στο τέλος του αλγορίθμου δεν υπάρχει κανένας κύκλος στον γράφο, και όλες οι ακμές είναι συνδεδεμένες μεταξύ τους, αφού από κάθε κύκλο έχει αφαιρεθεί μόνο μια ακμή, άρα όλες οι κορυφές του ενώνονται ακόμα. Επομένως το δέντρο είναι συνδετικό.

Είναι επίσης ελάχιστο. Έστω ακμή  $e$  που δεν ανήκει στο δέντρο και δημιουργεί κύκλο μαζί με αυτό. Αν η  $e$  δεν είναι η μεγαλύτερη ακμή του κύκλου, τότε υπάρχει ακμή στον κύκλο την οποία δεν διέγραψε ο αλγόριθμος, που δεν γίνεται. Επομένως, όλες οι ακμές που δεν περιλαμβάνονται στο δέντρο είναι οι μεγαλύτερες για του μεμονομένους κύκλους που δημιουργούνται, και άρα το συνδετικό δέντρο είναι ελάχιστο.

- (γ) Ταξινομούμε σε φθίνουσα σειρά μια λίστα με τις ακμές σε  $O(|E| \log |E|)$ .  
 Όσο υπάρχουν ακόμα ακμές στη λίστα:  
 Αφαίρεση μιας ακμής απο την λίστα με μέγιστο βάρος και εύρεση ενός κύκλου που περιλαμβάνει την συγκεκριμένη ακμή σε  $O(|E|)$   
 Αν βρέθηκε κύκλος, διαγραφή της ακμής σε  $O(1)$ .  
 Αν δεν βρέθηκε, έχουμε βρεί ήδη το ΕΣΔ.  
 Συνολικά,  $O(|E|^2)$ .