Αλγόριθμοι και Πολυπλοκότητα

Γαβαλάς Νίκος, ΑΜ 03113121

Δεκέμβριος 2018

3η Γραπτή Σειρά Ασκήσεων

Άσκηση 1

Έστω $T(V, E)$ δέντρο που κάθε κορυφή του u έχει βάρος $w(u) \in \mathbb{Z}$ και ζητάμε το μονοπάτι p στο δέντρο που μεγιστοποιεί το $\sum_{u \in p} w(u)$.

Η ιδέα είναι η εξής: Ορίζουμε αρχικά δύο ρίζες, η μία είναι η ρίζα του δέντρου αναζήτησης (επιλέγεται ως τυχαίος κόμβος κατά την εκκίνηση) και η άλλη η ρίζα-απάντηση, η ρίζα του υποδέντρου που αποτελεί το p.

Καλούμε την κάτωθι συνάρτηση αναδρομικά για κάθε κόμβο. Όταν τερματίσει, το μέγιστο άθροισμα είναι στη μεταβλητή $resVal$ και το p κατασκευάζεται με διάσχιση του υποδέντρου που εκκινεί από το $resNode$ ακολουθώντας τα $maxChild$.

Ο αλγόριθμος αυτός έχει γραμμική πολυπλοκότητα $O(|V| + |E|)$ καθώς δεν είναι παρά μια παραλλαγή της αναζήτησης κατά βάθος.

Algorithm 1 Max Sum Path

$resNode \leftarrow NULL$
$resVal \leftarrow -\infty$
$initNode \leftarrow$ chooseRandom()
function MAXSUM(node, parent)
  if $node == NULL$ **then return** 0
  for all child in node.adjList() **do**
    $listOfSums.append(maxSum(child, node))$
  $maxRight = max(listOfSums.max() + node.weight, node.weight)$ ▷ max if path continues to right
  $maxChild = ...$ ▷ set maxChild to the node which has the maxRight value
  $maxTop = max(maxRight, listOfSums.sum() + node.weight)$ ▷ max if path continues to top
  if maxTop > resVal **then**
    $resVal \leftarrow maxTop$
    $resNode \leftarrow node$
  return $maxRight, maxChild$
maxSum($initNode$, $NULL$)

Άσκηση 2

Έχουμε γράφο $G(V, E)$ με $p(u) \in \mathbb{N}, u \in V$ και ζητάμε την τιμή $c(u), \forall u \in V$, όπου $c(u)$ είναι η τιμή της φθηνότερης κορυφής που είναι προσπελάσιμη από τη u, μαζί με τη u.

α

Αν ο γράφος G είναι DAG, γνωρίζουμε πως τότε ορίζεται τοπολογική διάταξη μεταξύ των κορυφών του, η οποία σχηματίζεται σε χρόνο $O(|V| + |E|)$ (γραμμικό).

Διατάσσουμε λοιπόν τις κορυφές και ύστερα, προσπελαύνοντας τες αντίστροφα, κρατάμε για κάθε μία τη μικρότερη τιμή $c(u)$ βάσει της σχέσης $c(u) = \min\{p(u), \min_{v:(u,v)\in E}\{c(v)\}\}$.

Το δεύτερο βήμα απαιτεί επίσης γραμμικό χρόνο, επομένως ο αλγόριθμος είναι γραμμικού χρόνου.

β

Γνωρίζουμε ότι κάθε γράφος μπορεί να παρασταθεί ως DAG των SCC του, σε γραμμικό χρόνο.

Κάθε κορυφή $u \in W$, W SCC του G, θα έχει την ίδια τιμή $c(u)$ με τις υπόλοιπες κορυφές που ανήκουν στην ίδια SCC, αφού ακριβώς επειδή είναι ισχυρά συνεκτική συνιστώσα και άρα υπάρχει διαδρομή από και προς οποιοδήποτε $u, v \in W$, αρκεί να βρούμε την ελάχιστη $p(u)$ μεταξύ όλων αυτών και ύστερα σε όλες θα θέσουμε $c(u)$ ίση με αυτήν.

Αρχικά, βρίκουμε λοιπόν τις SCC, και για κάθε μία W από αυτές υπολογίζουμε την ελάχιστη τιμή μεταξύ όλων των κορυφών $u \in W$, δηλαδή το $P(W) = \min_{u\in W}\{p(u)\}$.

Στη συνέχεια, σχηματίζουμε τον μεταγράφο που αποτελεί το DAG των SCC W του G, και τρέχουμε τον αλγόριθμο του ερωτήματος (α) υπολογίζοντας τις τιμές $C(W)$ με βάση τα $P(W)$.

Τέλος, για κάθε SCC W, για κάθε κορυφή $u \in W$, θέτουμε $c(u) = C(W)$.

Η χρονική πολυπλοκότητα του αλγορίθμου είναι γραμμική, αφού κάθε βήμα απαιτεί χρόνο $O(|V| + |E|)$.

Άσκηση 4

Έστω συνεκτικός, μη κατευθυνόμενος γράφος $G(V, E)$ με $|V| = n, |E| = m$.

α

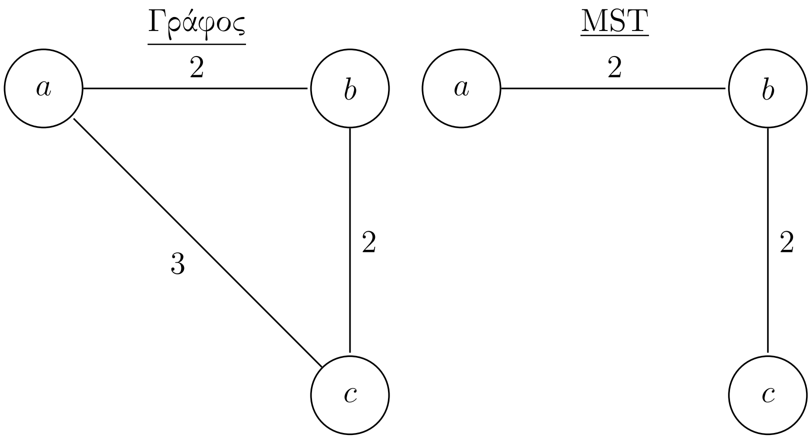
Έστω T_1, T_2 συνεκτικά δέντρα και ακμή $e' \in T_2 \backslash T_1$. Το $T_1 \cup \{e'\}$ θα περιέχει κύκλο C. Σε αυτόν τον κύκλο, $\exists e \in C : e \notin T_2$, αφού διαφορετικά το T_2 θα περιείχε τον C και δεν θα ήταν συνδετικό δέντρο. Το $(T_1\backslash\{e\}) \cup \{e'\}$ είναι άκυκλος γράφος με $|E| = |V| - 1$, άρα είναι συνδετικό δέντρο. Επομένως, $\forall e \in T_1\backslash T_2, \exists e' \in T_2\backslash T_1 : (T_1\backslash\{e\}) \cup \{e'\}$ είναι συνδ. δέντρο.

Μια τέτοια ακμή μπορεί να βρεθεί σε γραμμικό χρόνο ως εξής: για $e = (u, v) \in T_1\backslash T_2$, πραγματοποιούμε στο T_2 μια DFS αναζήτηση από το $u \to v$ και για κάθε ακμή e' που ανήκει στο προκύπτον μονοπάτι εξετάζουμε αν $\in T_1$ σε $(O(1))$ και επιστρέφουμε την πρώτη που $\notin T_1$. Η πολυπλοκότητα είναι αυτή του DFS, $O(|V|)$.

Άσκηση 5

Έστω $G(V, E, w)$ συνεκτικός και μη κατευθυνόμενος γράφος.

α



β

Αφού για κάθε τομή, η ακμή ελάχιστου βάρους που διασχίζει την τομή είναι μοναδική, τότε όλες οι ακμές θα έχουν διαφορετικά βάρη, γιατί αν υπήρχαν τουλάχιστον δύο ακμές με το ίδιο βάρος θα υπήρχε τομή για την οποία από τις διασχίζουσες ακμές είτε θα υπήρχε μικρότερη (οπότε δεν μας νοιάζει), ή θα ήταν και οι δύο οι μικρότερες (εφόσον είναι ίσες) και άρα δεν θα υπήρχε μοναδική ακμή ελάχιστου βάρους. Συνεπώς, γνωρίζοντας πως όταν σε συνδετικό μη κατευθυνόμενο γράφο όλες οι ακμές έχουν διαφορετικά βάρη, τότε το MST του γράφου είναι μοναδικό, προκύπτει το ζητούμενο.

Αντιπαράδειγμα: