Αλγοριθμική Επιστήμη Δεδομένων

2η Σειρά Ασκήσεων

Στοιχεία φοιτητή
Ονοματεπώνυμο: Κωνσταντίνος Τσόπελας
Αριθμός Μητρώου: 03400198

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Εθνικό Μετσόβιο Πολυτεχνείο
ΔΠΜΣ ΕΔΕΜΜ
3 Απριλίου 2024

Άσκηση 1

(α) Δεδομένου του ορισμού της καθολικότητας, αρκεί να δώσουμε ένα αντιπαράδειγμα ενός ζεύγους τιμών $x, y \in U = [m^k]$ για τις οποίες δίνουν την ίδια τιμή πάνω από $\mathcal{H}/m$ συναρτήσεις της οικογένειας.

Στην περίπτωσή μας, αρκεί να πάρουμε $x = 0, y = m$ (το m είναι μέσα, αφού $k \geq 2$). Τότε, έχουμε:

$$\begin{aligned} h_{a,b}(x = 0) &= (a0 + b) \mod m = b \mod m \\ h_{a,b}(y = m) &= (am + b) \mod m \\ &= ((am) \mod m + b \mod m) \mod m \\ &= (0 + b) \mod m = b \mod m \end{aligned}$$

Οπότε, **όλες** οι συναρτήσεις της οικογένειας δίνουν την ίδια τιμή για 0 και m, και άρα προφανώς δεν ισχύει η καθολικότητα.

(β), (γ) Και για τα δύο ερωτήματα, ισχύει αυτούσια η ανάλυση που κάναμε στο μάθημα για την ίδια οικογένεια συναρτήσεων, για $p > m$ (αφού $k \geq 2$) και $U = [p]$.

Οπότε, σε αυτή την περίπτωση, που είναι ακριβώς η περίπτωση του ερωτήματος (γ), γνωρίζουμε ήδη ότι ισχύει η καθολικότητα! (δείτε διαφάνειες "Hash-Table-Basics", σελίδα 17).

Επιπλέον, είναι προφανές από τον ορισμό ότι η καθολικότητα ισχύει πάντα και για υποσύνολα του σύμπαντος $U' \subset U$. Αν ισχύει η ιδιότητα για όλα τα ζεύγη του U, θα ισχύει και για όλα τα ζεύγη ενός υποσυνόλου του, καθώς ο ορισμός δεν εξαρτάται, δεν περιλαμβάνει κάπου το μέγεθος του U πχ. Οπότε, σίγουρα θα ισχύει και για την περίπτωση που περιγράφεται στο ερώτημα (β), αφού $p > m^k$.

Συνεπώς, και τα δύο ερωτήματα "ανάγονται" στο θεώρημα που δείξαμε στο μάθημα. Χάριν πληρότητας, θα δώσουμε εδώ μία πιο λεπτομερή απόδειξη του θεωρήματος αυτού.

Απόδειξη απλής καθολικής οικογένειας Αρχικά, επαναλαμβάνουμε το setting. Έχουμε ένα σύμπαν $U = [p]$, όπου p κάποιος μεγάλος πρώτος αριθμός, και εξετάζουμε την οικογένεια συναρτήσεων $h_{a,b}(x) = ((ax + b) \mod p) \mod m$ με τις παραμέτρους $a \in \mathbb{Z}_p^*, b \in \mathbb{Z}_p$. Ο m είναι κάποιος φυσικός αριθμός, μικρότερος του p.

Έστω δύο σημεία $x_1, x_2 \in \mathbb{Z}_p$ με $x_1 \neq x_2$. Έχοντας "φιξάρει" αυτούς τους δύο αριθμούς, θα δείξουμε, αρχικά, ότι η απεικόνιση:

$$\begin{aligned} (a, b) &\mapsto (y_1, y_2) \\ y_1 &= ax_1 + b \mod p \\ y_2 &= ax_2 + b \mod p \end{aligned} \tag{1}$$

είναι ένα προς ένα.

Πράγματι, έστω $y_1' = a'x_1 + b' \mod p$ και $y_2' = a'x_2 + b' \mod p$ και $y_1 = y_1'$, $y_2 = y_2'$. Τότε, λαμβάνουμε:

$$\begin{aligned} ax_1 + b \equiv a'x_1 + b' (\mod p) &\Leftrightarrow p|(a - a')x_1 + (b - b') \\ ax_2 + b \equiv a'x_2 + b' (\mod p) &\Leftrightarrow p|(a - a')x_2 + (b - b') \end{aligned}$$

Τότε, το p θα διαιρεί και τη διαφορά, δηλαδή:

$$p|(a-a')x_1+(b-b')-(a-a')x_2-(b-b')=(a-a')(x_1-x_2)$$

Και οι δύο όροι του γινομένου είναι μικρότεροι του p, και επειδή ο p είναι πρώτος, είναι αδύνατον να διαιρεί αυτό το γινόμενο, εκτός και αν αυτό είναι 0. Και αφού $x_1 \neq x_2$, αυτό συνεπάγεται ότι $a = a'$, και δεδομένου αυτού έυκολα βλέπει κανείς ότι πρέπει ο p να διαιρεί το $b - b'$, άρα και αυτά θα είναι ίσα.

Αυτό, λοιπόν, σημαίνει τώρα ότι η απεικόνιση (1) είναι τελείως αμφιμονοσήμαντη. Άρα, το πλήθος των ζευγών (a, b) για τα οποία η $h_{a,b}$ απεικονίζει τα x_1, x_2 στον ίδιο αριθμό (που είναι η ποσότητα που μας ενδιαφέρει) ισούται με το πλήθος των αριθμών $y_1, y_2 \in [p]$ που αποτελούν εικόνα αυτών των ζευγών (a, b) μέσω της (1).

Αυτά, όμως, τα ζεύγη (y_1, y_2), αφού αφορούν περιπτώσεις σύγκρουσης, έχουν όλα την ιδιότητα ότι τα y_1, y_2 έχουν το ίδιο υπόλοιπο με το m! Άρα, συμπεραίνουμε ότι το ζητούμενο πλήθος φράσσεται από:

$$|\{h_{a,b} : h_{a,b}(x_1) = h_{a,b}(x_2)\}| \leq |\{(y_1, y_2) \in [p]^2 : y_1 \equiv_m y_2\}|$$

Για να το φράξουμε τώρα αυτό το πλήθος, δεν μένει παρά να χρησιμοποιήσουμε "μετρητικά" επιχειρήματα. Συγκεκριμένα, για οποιοδήποτε φιξαρισμένο y_1, ας εξετάσουμε τα y_2 για τα οποία ισχύει $y_1 \equiv_m y_2$. Αυτό ισοδυναμεί με $m|y_1 - y_2$, άρα όλα αυτά τα y_2 θα είνα της μορφής:

$$y_2 = y_1 + \kappa m, \kappa \in \mathbb{Z}$$

Συνδυάζοντας την σχέση αυτή με το ότι το y_2 είναι ένας αριθμός στο $[p]$, θα έχουμε ότι πρέπει να ισχύει:

$$0 \leq y_1 + \kappa m, \kappa \in \mathbb{Z} \leq p-1 \Leftrightarrow -\frac{y_1}{m} \leq \kappa \leq \frac{p-1}{m} - \frac{y_1}{m}$$

Είναι, λοιπόν, φανερό ότι το πλήθος των ακεραίων κ για τους οποίους ισχύει το παραπάνω ταυτίζεται, καταρχάς, με το πλήθος των ακεραίων στο διάστημα $[0, \frac{p-1}{m} = \frac{p}{m} - \frac{1}{m}]$, το οποίο, βέβαια, δεν μπορεί να είναι περισσότερο από:

$$\left\lceil \frac{p}{m} \right\rceil - 1$$

Αν, λοιπόν, επανέλθουμε τώρα στην καταμέτρηση των ζευγών y_1, y_2, αφού $y_1 \in [p]$, αυτά δεν μπορούν να είναι περισσότερα από:

$$p\left(\left\lceil \frac{p}{m} \right\rceil - 1\right)$$

Τελικά, το πλήθος όλων των ζευγών (a, b) ή (y_1, y_2) (το ίδιο είναι) άρα και των συναρτήσεων της οικογένειας, είναι $p(p-1)$. Άρα, η πιθανότητα, εν τέλει, να δοθεί η ίδια τιμή στα x_1, x_2 είναι το πολύ ίση με:

$$\frac{p\left(\left\lceil \frac{p}{m} \right\rceil - 1\right)}{p(p-1)} = \frac{\left\lceil \frac{p}{m} \right\rceil - 1}{p-1} \leq \frac{\frac{p+m-1}{m} - 1}{p-1} = \frac{1}{m}$$

Άσκηση 2

(α) Επιτυχής αναζήτηση κλειδιού k: Δοκιμή με τη σειρά όλων των θέσεων $h(k,i), i = 0,\ldots,m-1$ μέχρι την πρώτη στην οποία βρίσκουμε το κλειδί k στην αντίστοιχη θέση του πίνακα (αφού είναι επιτυχής αναζήτηση).

Εισαγωγή κλειδιού k: Δοκιμή με τη σειρά όλων των θέσεων $h(k,i), i = 0,\ldots,m-1$ μέχρι την πρώτη στην οποία βρίσκουμε κενή την αντίστοιχη θέση του πίνακα.

Από την παραπάνω περιγραφή, βλέπουμε ότι, μετά από εισαγωγή n στοιχείων, και οι δύο διαδικασίες κάνουν, ουσιαστικά, το ίδιο πράγμα! Δηλαδή, περνούν από τις ίδιες θέσεις του πίνακα, και σταματούν μετά από το ίδιο πλήθος προσπελάσεων.

Συνεπώς, οι χρόνοι που απαιτούν οι δύο παραπάνω διαδικασίες θα είναι, προφανώς, ίδιοι!

(β) Ξεκινάμε με την εκτίμηση που μας προτείνεται από την υπόδειξη, του αναμενόμενου πλήθους δοκιμών κατά την εισαγωγή του i-οστού στοιχείου.

Υποθέτοντας uniform hashing, θεωρούμε ότι η δοκιμή κάθε θέσης $h(k,j)$ του πίνακα θα έχει πιθανότητα i/m να είναι κατειλημμένη και $1-i/m$ να μην είναι (αφού έχουν καταληφθεί, μέχρι στιγμής, i θέσεις, υποθέτοντας ότι η αρίθμησή μας ξεκινάει από το 0), ανεξάρτητα κάθε φορά από τις υπόλοιπες δοκιμές.

Κατά συνέπεια, η διαδικασία εισαγωγής του i-οστού στοιχείου μοντελοποιείται, πιθανοτικά, από μία σειρά ανεξάρτητων δοκιμών Bernoulli με πιθανότητα "επιτυχίας" $p = 1 - \frac{i}{m}$. Το αναμενόμενο πλήθος δοκιμών ταυτίζεται, τότε, με το αναμενόμενο πλήθος δοκιμών μέχρι την πρώτη επιτυχία, που είναι η μέση τιμή της γεωμετρικής κατανομής, και είναι ίση με:

$$\frac{1}{p} = \frac{1}{1-\frac{i}{m}} = \frac{m}{m-i}$$

Ισχυριζόμαστε, τώρα, το εξής. Ο ζητούμενος μέσος χρόνος, που όπως είπαμε στο ερώτημα (α) είναι και ο μέσος χρόνος εισαγωγής στοιχείου, μπορεί να υπολογιστεί, με μεγαλύτερη ακριβεια, ως ο μέσος όρος των χρόνων εισαγωγής κάθε ενός από τα n στοιχεία που βάζουμε ένα ένα στον πίνακα!

Οπότε, θα έχουμε:

$$\frac{1}{n}\sum_{i=0}^{n-1}\frac{m}{m-i} = \frac{m}{n}\sum_{i=0}^{n-1}\frac{1}{m-i} = \frac{m}{n}\sum_{i'=m-n+1}^{m}\frac{1}{i'}$$

Σε αυτό το σημείο, μπορούμε να χρησιμοποιήσουμε την κλασσική τεχνική η οποία χρησιμοποιείται π.χ. για κάποια έυκολα λογαριθμικά φράγματα της αρμονικής σειράς. Συγκεκριμένα, ότι το παραπάνω άθροισμα μπορούμε να το σκεφτούμε ως ολοκλήρωμα μιας step function που ξεκινάει από το $m-n>0$, φτάνει μέχρι το m και σε κάθε διάστημα παίρνει την τιμή της ακολουθίας στο δεξί άκρο του διαστήματος.

Εύκολα μπορεί κανείς να δει ότι αυτή η συνάρτηση είναι ολόκληρη κάτω από τον λογάριθμο. Και άρα, τελικά, ο ζητούμενος μέσος χρόνος θα φράσσεται από την ποσότητα:

$$\frac{m}{n}\int_{x=m-n}^{m}\frac{\mathrm{d}x}{x} = \frac{1}{n/m}(\ln m - \ln(m-n)) = \frac{1}{\alpha}\ln\frac{1}{\frac{m-n}{m}} = \frac{1}{\alpha}\ln\frac{1}{1-\frac{n}{m}} = \frac{1}{\alpha}\ln\frac{1}{1-\alpha}$$

Άσκηση 3

(α) Καταρχάς, ο αλγόριθμος Girvan-Newman εκτελεί την ίδια ακριβώς διαδικασία μία φορά για κάθε μία κορυφή ως source, και μετά για κάθε ακμή απλά προσθέτει τα επιμέρους αποτελέσματα. Οπότε, εδώ, θεωρούμε ότι αρκεί να δείξουμε σε μορφή ψευδοκώδικα τη διαδικασία που επιτελείται για μία κορυφή ως πηγή, έστω ότι την ονομάζουμε U.

Ο αλγόριθμος εξελίσσεται σε δύο φάσεις. Θα τις περιγράψουμε ξεχωριστά.

Η πρώτη φάση είναι ένας απλός BFS για τον υπολογισμό του πλήθους των ελαχίστων μονοπατιών από την κορυφή U προς κάθε μία από τις υπόλοιπες κορυφές. Η περιγραφή του σε ψευδοκώδικα φαίνεται στον Αλγόριθμο 1.

Algorithm 1 Πρώτη φάση Girvan-Newman

$Q \leftarrow$ μια ουρά (FIFO) που περιέχει μόνο την κορυφή U
$S \leftarrow$ μια άδεια αρχικά στοίβα (LIFO)
$State \leftarrow$ πίνακας με την κατάσταση κάθε κορυφής: A = ανέγγιχτη, E = έχει αγγιχτεί. Αρχικά όλες A.
$Paths \leftarrow$ πίνακας με τα πλήθη ελαχίστων μονοπατιών για κάθε κορυφή. Αρχικά, για όλες άγνωστο, πλην της U για την οποία είναι 1.
$Depths \leftarrow$ πίνακας με το βάθος (i.e. ελάχιστη απόσταση από την U) για κάθε κορυφή. Αρχικά για όλες άγνωστο, πλην της U για την οποία είναι 0.
while Q δεν είναι άδεια **do**
    $u \leftarrow Q.pop()$
    $S.push(u)$
    for όλους τους γείτονες v της u **do**
        if State[v] = A **then**
            $Depths[v] \leftarrow Depths[u] + 1$
            $State[v] \leftarrow$ E
            $Paths[v] \leftarrow Paths[u]$
            $Q.add(v)$
        else if $Depths[v] = Depths[u] + 1$ **then**
            $Paths[v] \leftarrow Paths[v] + Paths[u]$
        end if
    end for
end while

Στο τέλος της φάσης αυτής, 3 πληροφορίες κρατάμε:

- Τον πίνακα $Paths$ με τα πλήθη των ελαχίστων μονοπατιών για όλες τις κορυφές.
- Τη στοίβα S στην οποία έχουμε κρατήσει τις κορυφές σε αντίστροφη σειρά από αυτήν που τις επισκεφτήκαμε στον BFS (πλην της κορυφής U, αλλά δεν έχει ιδιαίτερη σημασία).
- Τον πίνακα $Depths$ που έχει, για κάθε κορυφή, το βάθος της στο BFS δέντρο (ισοδύναμα, την απόστασή της από την κορυφή U).

Στην συνέχεια, η δεύτερη φάση είναι ακόμη μία διάσχιση όλων των κορυφών, από κάτω προς τα πάνω, για να υπολογίσουμε, τελικά, για κάθε ακμή το άθροισμα των

ποσοστών των ελαχίστων μονοπατιών από την U προς οποιαδήποτε άλλη κορυφή που περνάνε από την ακμή.

Η διαδικασία είναι όπως ακριβώς τη συζητήσαμε στο μάθημα, όπου εννοιολογικά αναθέτουμε σε κάθε κορυφή μία μονάδα "ροής", την οποία η κορυφή μοιράζει στους προγόνους της σύμφωνα με τα πλήθη ελαχίστων μονοπατιών που έχει ο καθένας.

Ο ψευδοκώδικας για την φάση αυτή παρουσιάζεται στον Αλγόριθμο 2.

Algorithm 2 Δεύτερη φάση Girvan-Newman

$BetweennessU \leftarrow$ πίνακας στον οποίο θα αποθηκεύσουμε, για κάθε ακμή, το τελικό αποτέλεσμα, δηλαδή το edge betweenness λαμβάνοντας υπόψη μόνο μονοπάτια που ξεκινούν από την U.
$PathFraction \leftarrow$ πίνακας που θα περιέχει, για κάθε κορυφή, τον (κλασματικό) αριθμό μονοπατιών που "διαθέτει" για να "διανείμει" στους προγόνους της. Τον αρχικοποιούμε σε 1 για όλες τις κορυφές.
$Depths \leftarrow$ πίνακας με το βάθος (i.e. ελάχιστη απόσταση από την U) για κάθε κορυφή. Αρχικά για όλες άγνωστο, πλην της U για την οποία είναι 0.
while S δεν είναι άδεια **do**
  $u \leftarrow S.pop()$
  for όλους τους γείτονες v της u **do**
    if Depths[v] = Depths[u] - 1 **then**
      $pathfraction \leftarrow \frac{Paths[v]}{Paths[u]} \cdot PathFraction[u]$
      $BetweennessU[\{u, v\}] = pathfraction$
      $PathFraction[v] \leftarrow PathFraction[v] + pathfraction$
    end if
  end for
end while

Τέλος, να σημειώσουμε ότι κάθε edge betweenness πρέπει στο τέλος να διαιρεθεί με το 2, καθώς κάθε κλάσμα ελαχίστων μονοπατιών μεταξύ κάθε ζεύγους κορυφών έστω u, v έχει καταμετρηθεί ακριβώς 2 φορές, μία όταν πήραμε την u ως source και μία όταν πήραμε την v ως source.

Όσον αφορά την πολυπλοκότητα, κάθε μία από τις παραπάνω φάσεις είναι ουσιαστικά πολύ κοντά σε έναν BFS, συγκεκριμένα κάθε ακμή προσπελαύνεται το πολύ 2 φορές, μία από το κάθε άκρο της. Οπότε, η πολυπλοκότητα των δύο φάσεων είναι γραμμική, $O(|V| + |E|)$.

Αυτές οι 2 φάσεις τρέχουν μία φορά για κάθε κορυφή, άρα η συνολική πολυπλοκότητα του αλγορίθμου είναι $O(|V|^2 + |V||E|)$.

(β) Στην περίπτωση των δέντρων, ο αλγόριθμος μπορεί να απλοποιηθεί αρκετά κάνοντας ορισμένες παρατηρήσεις.

Καταρχάς, σε ένα δέντρο, κάθε ζεύγος κορυφών u, v συνδέεται με μοναδικό (απλό) μονοπάτι (αλλιώς θα είχαμε κύκλο), και άρα και μοναδικό ελάχιστο μονοπάτι. Οπότε, μία ακμή είτε θα ανήκει σε αυτό το μονοπάτι και το edge betweenness της θα παίρνει $+1$ από αυτό το ζεύγος ακμών, είτε δεν θα ανήκει και δεν θα παίρνει τίποτα από αυτό το ζεύγος ακμών. Με άλλα λόγια, το edge betweenness στην περίπτωση των δέντρων ταυτίζεται απλά με το πλήθος των (απλών) μονοπατιών που περνούν από την ακμή.

Από τη σκοπιά των ακμών, τώρα, κάθε ακμή e είναι γέφυρα του δέντρου, την οποία, αν αφαιρέσουμε, προκύπτουν δύο ξένα υπόδεντρα. Κατόπιν τούτου, για κάθε

ζεύγος κορυφών u, v, είτε αυτές ανήκουν στο ίδιο υπόδεντρο, ένα εκ των δύο, είτε η μία ανήκει στο ένα και η άλλη στο άλλο.

Είναι εμφανές ότι στην πρώτη περίπτωση, το μονοπάτι $u-v$ δεν μπορεί να περιέχει την ακμή e, καθώς αν την περάσουμε μία φορά, θα πρέπει να την ξαναπεράσουμε για να επιστρέψουμε στο ίδιο υπόδεντρο, και αυτό δεν είναι απλό, και άρα ούτε και ελάχιστο, φυσικά, μονοπάτι.

Στη δεύτερη, περίπτωση, αντίθετα, είναι σίγουρο ότι η e θα ανήκει στο μονοπάτι $u - v$, αφού δεν υπάρχει άλλος τρόπος να μετακινηθούμε από το ένα υπόδεντρο στο άλλο εκτός από την e.

Συμπεραίνουμε, λοιπόν, ότι τα μονοπάτια στα οποία ανήκει μία ακμή e είναι ακριβώς όλα τα μονοπάτια $u-v$, όπου η u ανήκει στο υπόδεντρο T_1 και η v στο υπόδεντρο T_2, όπου T_1, T_2 είναι τα υπόδεντρα που προκύπτουν με την αφαίρεση της e. Κατά συνέπεια, το πλήθος των μονοπατιών αυτών ισούται με $|V(T_1)| \cdot |V(T_2)|$.

Κάπως έτσι, καταλήγουμε στον αλγόριθμο που πρέπει να ακολουθήσουμε, ο οποίος αρκεί, αφού διαλέξουμε μια οποιαδήποτε κορυφή για ρίζα του δέντρου, να κάνει μία απλή διάσχιση του δέντρου (BFS ή DFS) για να υπολογίσει το πλήθος των κόμβων κάθε υπόδεντρου. Μετά, το edge betweenness προκύπτει με έναν απλό πολλαπλασιασμό.

Ο αλγόριθμος παρουσιάζεται, χωρίς πολλές λεπτομέρειες, στο σχήμα 3.

Algorithm 3 Edge Betweenness σε Δέντρα

$r \leftarrow$ μία οποιαδήποτε ακμή του δέντρου που επιλέγουμε ως ρίζα.
$parent, children \leftarrow$ ο γονιός και η λίστα με τα παιδιά του κάθε κόμβου. Υπολογίζεται εύκολα με π.χ. έναν BFS στην αρχική αναπαράσταση του γράφου.
$nodes \leftarrow$ πίνακας όπου θα αποθηκεύσουμε το πλήθος των κορυφών κάθε υπόδεντρου, αρχικά κενός.
for όλες τις κορυφές u, αναδρομικά, ξεκινώντας με την r **do**
    if children[u] κενή (δηλαδή είναι φύλλο) **then**
        $nodes[u] \leftarrow 1$
    else
        $nodes[u] \leftarrow \sum_{v \in children[u]}$ (calculate number of nodes of v, plus save it in $nodes[v]$)
    end if
end for
for $u \in V$ **do**
    $BetweennessU[\{u, parent[u]\}] \leftarrow nodes[u] \cdot (|V| - nodes[u])$
end for

Ο αλγόριθμος αυτός είναι προφανώς γραμμικού χρόνου, αφού κάνει μία διάσχιση του δέντρου για την εύρεση του πλήθους κορυφών και μία ακόμα διάσχιση για τον υπολογισμό του betweenness.

Άσκηση 4

MMDS, Άσκηση 10.4.1

(a) Ο πίνακας γειτνίασης για τον γράφο του σχήματος 10.9 είναι ο ακόλουθος:

$$A = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \\ G \\ H \\ I \end{array} \begin{array}{c} \begin{array}{ccccccccc} A & B & C & D & E & F & G & H & I \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix} \end{array}$$

(b) Ο διαγώνιος πίνακας βαθμών είναι:

$$D = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \\ G \\ H \\ I \end{array} \begin{array}{c} \begin{array}{ccccccccc} A & B & C & D & E & F & G & H & I \end{array} \\ \begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{pmatrix} \end{array}$$

Τέλος, ο Laplacian δεν είναι παρά η διαφορά τους:

$$L = D - A = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \\ G \\ H \\ I \end{array} \begin{array}{c} \begin{array}{ccccccccc} A & B & C & D & E & F & G & H & I \end{array} \\ \begin{pmatrix} 2 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & 0 & 0 & 0 & -1 & 0 \\ -1 & -1 & 3 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 3 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 3 & -1 & -1 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 3 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 2 \end{pmatrix} \end{array}$$

MMDS, Άσκηση 10.4.2

Κατόπιν αρκετής διερεύνησης, δεν καταφέραμε να βρούμε κάποιον "έξυπνο" τρόπο για τον υπολογισμό της δεύτερης μικρότερης ιδιοτιμής του λαπλασιανού. Συγκεκριμένα, δοκιμάσαμε να προχωρήσουμε αναλυτικά την ελαχιστοποίηση του $x^T Lx$ (που αναφέρει το βιβλίο) με τις συνθήκες ΚΚΤ, αλλά δεν καταλήξαμε σε κάποιο χρήσιμο

```python
import numpy as np

L = np.array([
    [ 2, -1, -1,  0,  0,  0,  0,  0,  0],
    [-1,  3, -1,  0,  0,  0,  0, -1,  0],
    [-1, -1,  3, -1,  0,  0,  0,  0,  0],
    [ 0,  0, -1,  3, -1, -1,  0,  0,  0],
    [ 0,  0,  0, -1,  3, -1, -1,  0,  0],
    [ 0,  0,  0, -1, -1,  2,  0,  0,  0],
    [ 0,  0,  0,  0, -1,  0,  3, -1, -1],
    [ 0, -1,  0,  0,  0,  0, -1,  3, -1],
    [ 0,  0,  0,  0,  0,  0, -1, -1,  2]
])

values, vectors = np.linalg.eigh(L)

print(values)
print()
print(vectors.T)
```

Σχήμα 1: Κώδικας για τον υπολογισμό των ιδιοτιμών και των ιδιοδιανυσμάτων του πίνακα L

συμπέρασμα. Το επόμενο βήμα θα ήταν ίσως να δοκιμάσουμε να υπολογίσουμε το χαρακτηριστικό πολυώνυμο του L, το οποίο ίσως να έβγαζε κάπου, καθώς ο πίνακας είναι αρκετά αραιός και θα μπορούσαμε να κάνουμε πράξεις με το χέρι.

Ωστόσο, συνειδητοποιήσαμε ότι, από τη στιγμή που για το cut μας ενδιαφέρουν μόνο τα πρόσημα των συνιστωσών του ιδιοδιανύσματος, μπορούμε να βρούμε την δεύτερη μικρότερη ιδιοτιμή και το αντίστοιχο ιδιοδιάνυσμα υπολογιστικά και, αν οι συνιστώσες είναι αρκετά μακριά από το 0, τότε μπορούμε να πούμε με αρκετά μεγάλη σιγουριά ότι το πρόσημό τους δεν θα έχει επηρεαστεί από την περιορισμένη ακρίβεια των υπολογισμών.

Κατόπιν τούτου, χρησιμοποιούμε τον κώδικα python που φαίνεται στο σχήμα 1, για να υπολογίσουμε τις ιδιοτιμές και τα ιδιοδιανύσματα του πίνακα L.

Αγνοώντας την πρώτη ιδιοτιμή, που είναι μία τιμή πολύ κοντά στο 0, η δεύτερη μικρότερη ιδιοτιμή είναι περίπου 6.97224, και εμφανίζεται 2 φορές, δηλαδή έχει αλγεβρική πολλαπλότητα 2. Τα δύο αντίστοιχα ιδιοδιανύσματα είναι:

$$x_1 = (-0.1670, -0.2222, 0.0046, 0.3999, 0.3447, 0.5716, -0.1777, -0.3493, -0.4045)$$
$$x_2 = (-0.5636, -0.3335, -0.4007, -0.0257, 0.2044, 0.1371, 0.3592, 0.1963, 0.4264)$$

Οπότε, η διαμέριση που προκύπτει από το x_1 είναι στα σύνολα κορυφών $\{A, B, G, H, I\}$ και $\{C, D, E, F\}$.

Η διαμέριση που προκύπτει από το x_2 είναι στα $\{A, B, C, D\}$ και $\{E, F, G, H, I\}$.

Άσκηση 5

(a) Όπως γνωρίζουμε από τη θεωρία, η τιμή της normalized cut για μία διαμέριση S, T των κορυφών του γράφου ορίζεται ως:

$$\frac{Cut(S,T)}{Vol(S)} + \frac{Cut(S,T)}{Vol(T)}$$

όπου $Cut(S,T)$ είναι το πλήθος των ακμών που διασχίζουν την τομή και $Vol(S), Vol(T)$ είναι το πλήθος των ακμών όλου του γραφήματος που έχουν τουλάχιστον ένα άκρο τους στο S και το T, αντίστοιχα.

Για τον γράφο του σχήματος 10.16 και τη διαμέριση {1, 2, 3}, {4, 5, 6}, έχουμε:

1. $Cut(S,T) = 2$, οι ακμές που διασχίζουν είναι μόνο οι $(1,4)$ και $(3,6)$.
2. $Vol(S) = Vol(T)$ (λόγω συμμετρίας) $= 5$ (οι τρεις ακμές που είναι εντός της κάθε ομάδας και οι δύο που διασχίζουν).

Άρα, η τιμή της normalized cut θα είναι:

$$\frac{2}{5} + \frac{2}{5} = \frac{4}{5}$$

(b) Επαναλαμβάνουμε αρχικά τον ορισμό του modularity:

$$Q(G,S) = \frac{1}{2m} \sum_{s \in S} \sum_{i,j \in s} (A_{ij} - \frac{k_i k_j}{2m})$$

όπου:

1. S συμβολίζει τη διαμέριση των κόμβων, δηλαδή $s \in S$ είναι οι ομαδούλες.
2. m είναι το πλήθος των ακμών του γράφου.
3. k_i είναι ο βαθμός της κορυφής i.

Με βάση τον ορισμό, λοιπόν, για τη διαμέριση $S =$ { {1, 2, 3}, {4, 5, 6} } θα έχουμε:

$$\begin{aligned} Q(G,S) &= \frac{1}{2 \cdot 8} \left(\sum_{i,j \in \{1,2,3\}} (A_{ij} - \frac{k_i k_j}{2 \cdot 8}) + \sum_{i,j \in \{4,5,6\}} (A_{ij} - \frac{k_i k_j}{2 \cdot 8}) \right) \\ &= \frac{1}{16} \left(2(1 - \frac{2 \cdot 3}{16} + 1 - \frac{3 \cdot 3}{16} + 1 - \frac{2 \cdot 3}{16}) - \frac{3^2}{16} - \frac{2^2}{16} - \frac{3^2}{16} \right) \\ &+ (\text{το ίδιο, λόγω συμμετρίας των δύο ομάδων}) \\ &= \frac{2}{16} \left(2(3 - \frac{21}{16}) - \frac{22}{16} \right) = \frac{1}{8} \left(2\frac{27}{16} - \frac{11}{8} \right) = \frac{1}{8}\frac{16}{8} = \frac{1}{4} \end{aligned}$$

Στο παραπάνω, έχουμε αξιοποιήσει το γεγονός ότι οι δύο ομάδες είναι τελείως συμμετρικές, οπότε ο όρος που αφορά τη μία θα είναι ίσος με της άλλης, καθώς και ότι στο άθροισμα $\sum_{i,j \in \{1,2,3\}}$ κάθε ζεύγος διαφορετικών κόμβων εμφανίζεται δύο φορές, εξ ου και ο πολλαπλασιασμός επί 2.

Παρομοίως, υπολογίζουμε το modularity και για τη διαμέριση $S = \{ \{1\}, \{2, 3\}, \{4, 5\}, \{6\} \}$:

$$\begin{aligned}
Q(G,S) &= \frac{1}{16}\left(\sum_{i,j\in\{1\}}(A_{ij}-\frac{k_ik_j}{16})+\sum_{i,j\in\{2,3\}}(A_{ij}-\frac{k_ik_j}{16})\right.\\
&\quad\left.+\sum_{i,j\in\{4,5\}}(A_{ij}-\frac{k_ik_j}{16})+\sum_{i,j\in\{6\}}(A_{ij}-\frac{k_ik_j}{16})\right)+\\
&= \frac{1}{16}\left(-\frac{3^2}{16}+2(1-\frac{2\cdot 3}{16})-\frac{2^2}{16}-\frac{3^2}{16}+2(1-\frac{3\cdot 2}{16})-\frac{3^2}{16}-\frac{2^2}{16}-\frac{3^2}{16}\right)\\
&= \frac{1}{16}\left(-4\cdot\frac{9}{16}-2\cdot\frac{4}{16}+4(1-\frac{6}{16}\right)\\
&= \frac{1}{16}\left(-\frac{9}{4}-\frac{1}{2}+4\cdot\frac{5}{8}\right)\\
&= \frac{1}{16}(2-\frac{9}{4}) = -\frac{1}{16}\frac{1}{4} = -\frac{1}{64}
\end{aligned}$$

Άσκηση 6

MMDS, Άσκηση 8.3.1

(a) Ένα perfect matching για τον G_4 είναι το:

$$\{ (a_0, b_0), (a_1, b_2), (a_2, b_1), (a_3, b_3) \}$$

(b) Ένα perfect matching για τον G_5 είναι το:

$$\{ (a_0, b_0), (a_1, b_2), (a_2, b_4), (a_3, b_1), (a_4, b_3) \}$$

(c) Θα δείξουμε ότι κάθε G_n έχει perfect matching κατασκευαστικά. Διακρίνουμε δύο περιπτώσεις.

n **άρτιος** δηλαδή $n = 2k$. Τότε, ταιριάζουμε τις κορυφές ως εξής:

1. για $i = 0, \ldots, k-1$, παίρνω την ακμή $a_i - b_{2i}$. Παρατηρείστε ότι εδώ $2i < n$, εξ ου και έχει φύγει το modulo.

2. για $i = k, \ldots, 2k-1 = n-1$, παίρνω την ακμή $a_i - b_{(2i+1) \mod n}$.

Αρκεί να δείξουμε ότι στις κορυφές b που έχουμε πάρει στη μία και στην άλλη περίπτωση δεν υπάρχει καμία που να εμφανίζεται πάνω από μία φορά.

Θα το κάνουμε απαριθμώντας τα σύνολα των κορυφών b που εμπλέκονται σε κάθε περίπτωση:

1. Συνδέουμε τις a_i με τις $b_{2i}, i = 0, \ldots, k-1$. Δηλαδή, "δεσμεύονται" οι κορυφές b με τους δείκτες $\{2i, i = 0, \ldots, k-1\} = \{0, 2, 4, \ldots, 2(k-1) = n-2\}$.

2. Συνδέουμε τις a_i με τις $b_{(2i+1) \mod n}, i = k, \ldots, 2k-1$. Εδώ ισχύει:

$$\begin{aligned} &2i+1 = 2(k+i')+1, i' = 0, 1, \ldots, k-1 \\ =&2k+2i'+1 = n+2i'+1 \equiv_n 2i'+1, i' = 0, 1, \ldots, k-1 \end{aligned}$$

Οπότε, δεσμεύονται οι b με δείκτες $\{1, 3, 5, \ldots, 2(k-1)+1 = n-1\}$.

Είναι εμφανές ότι κανένας δείκτης κορυφής b δεν εμφανίζεται δύο φορές, οπότε όντως βρήκαμε ένα perfect matching.

n **περιττός** δηλαδή $n = 2k+1$. Τότε, ταιριάζουμε τις κορυφές ως εξής:

1. για $i = 0, \ldots, k-1$, παίρνω την ακμή $a_i - b_{2i}$ (όμοια με πριν).
2. για $i = k+1, \ldots, 2k = n-1$, παίρνω την ακμή $a_i - b_{(2i) \mod n}$.
3. τέλος, παίρνουμε και την ακμή (a_k, b_{n-1}).

Προχωράμε με την ίδια λογική όπως και στην πρώτη περίπτωση, και έχουμε:

1. Συνδέουμε τις a_i με τις $b_{2i}, i = 0, \ldots, k-1$. Όμοια με πριν, δεσμεύονται οι κορυφές b με τους δείκτες $\{2i, i = 0, \ldots, k-1\} = \{0, 2, 4, \ldots, 2(k-1) = n-3\}$.
2. Συνδέουμε τις a_i με τις $b_{(2i) \mod n}, i = k+1, \ldots, 2k$. Εδώ ισχύει:

$$\begin{aligned} &2i = 2(k+i'+1), i' = 0, 1, \ldots, k-1 \\ =&2k+2i'+2 = n-1+2i'+2 = n+2i'+1 \equiv_n 2i'+1, i' = 0, 1, \ldots, k-1 \end{aligned}$$

Οπότε, δεσμεύονται οι b με δείκτες $\{1, 3, 5, \ldots, 2(k-1)+1 = 2k-1 = n-2\}$.

3. Από τις δύο παραπάνω περιπτώσεις, είναι σαφές ότι το b_{n-1} μένει αδέσμευτο, άρα μπορούμε να το ταιριάξουμε με το a_k.

MMDS, Άσκηση 8.4.1

(a) Μπορούμε να το δούμε με τον εξής απλό τρόπο: οι B, C έχουν συνολικό budget 4 και έχουν κάνει bid και στα δύο queries x, y. Άρα, το πρώτο κομμάτι της ακολουθίας, το $xxyy$ σίγουρα μπορεί, αν μη τι άλλο, να ανατεθεί σε αυτούς τους δύο (με οποιαδήποτε σειρά - συνδυασμό).

Οπότε, τα 4 πρώτα queries σίγουρα θα ανατεθούν.

(b) Το απλούστερο θεωρώ ότι είναι να πάρουμε την ακολουθία $yyzz$. Δεδομένου ότι δεν εμφανίζεται καθόλου το x και κατά τα άλλα μόνο ο B κάνει bid στο y και ο C στα y, z, στην πραγματικότητα η περίπτωση αυτή είναι τελείως "ισόμορφη" με το παράδειγμα 8.7 του βιβλίου.

Οπότε (για πληρότητα), αυτό που συμβαίνει είναι ότι ο optimum offline αλγόριθμος θα ανέθετε τα yy στον B και τα zz στον C, αλλά ο greedy μπορεί, στην χειρότερη περίπτωση, να αναθέσει τα yy στον C, και μετά τα zz να μην μπορούν να ανατεθούν σε κανέναν, αφού το B δεν κάνει bid σε αυτά και ο C έχει εξαντλήσει το budget του.

Άσκηση 7

MMDS, Άσκηση 9.3.1

Ξαναγράφουμε τον αρχικό πίνακα ωφέλειας, που είναι το "ground truth" μας (1).

	a	b	c	d	e	f	g	h
A	4	5		5	1		3	2
B		3	4	3	1	2	1	
C	2		1	3		4	5	3

Πίνακας 1: Πίνακας ωφέλειας

(a) Θεωρώντας τον πίνακα ωφέλειας ως boolean (μη κενό $\to 1$, κενό $\to 0$), θα γίνει όπως φαίνεται στον Πίνακα 2.

	a	b	c	d	e	f	g	h
A	1	1	0	1	1	0	1	1
B	0	1	1	1	1	1	1	0
C	1	0	1	1	0	1	1	1

Πίνακας 2: Πίνακας ωφέλειας ως boolean

Με βάση τον πίνακα αυτό, υπολογίζουμε την απόσταση Jaccard μεταξύ κάθε ζεύγους χρηστών i, j κατά τον γνωστό τρόπο:

$$1 - \frac{\{\text{items του } i\} \cap \{\text{items του } j\}}{\{\text{items του } i\} \cup \{\text{items του } j\}}$$

Παρουζιάζουμε τα τελικά αποτελέσματα στον πίνακα 3.

	B	C
A	1/2	1/2
B	-	1/2

Πίνακας 3: Jaccard distance μεταξύ ζευγών χρηστών

(b) Δουλεύουμε με παρόμοιο τρόπο, απλά εδώ υπολογίζουμε την cosine distance μεταξύ των χρηστών i, j, με τον τύπο:

$$1 - \frac{(\text{διάνυσμα } i) \cdot (\text{διάνυσμα } j)}{\|\text{διάνυσμα } i\|_2 \cdot \|\text{διάνυσμα } j\|_2}$$

Υπολογίζουμε αρχικά τις νόρμες:

$$\|v_A\| = \sqrt{6}$$
$$\|v_B\| = \sqrt{6}$$
$$\|v_C\| = \sqrt{6}$$

και τα εσωτερικά γινόμενα, τα οποία αφού ο πίνακας έχει γίνει boolean, δεν είναι τίποτα άλλο από το μέγεθος της τομής, το οποίο υπολογίσαμε και στο προηγούμενο ερώτημα:

$$v_A \cdot v_B = 4$$
$$v_A \cdot v_C = 4$$
$$v_B \cdot v_C = 4$$

Από τη στιγμή που όλα τα μεγέθη ταυτίζονται, όλες οι cosine distances θα είναι ίσες με:

$$1 - \frac{4}{\sqrt{6}\sqrt{6}} = 1 - \frac{4}{6} = 1 - \frac{2}{3} = \frac{1}{3}$$

Οι τελικές cosine distances, παρόλα αυτά, φαίνονται αναλυτικά στον πίνακα 4.

	B	C
A	$1/3$	$1/3$
B	-	$1/3$

Πίνακας 4: Cosine distance μεταξύ ζευγών χρηστών

(c) Κάνοντας την αντιστοίχιση 3, 4, 5 $\rightarrow$ 1 και 1, 2, κενό $\rightarrow$ 0, ο πίνακας ωφέλειας γίνεται όπως φαίνεται στον 5.

	a	b	c	d	e	f	g	h
A	1	1	0	1	0	0	1	0
B	0	1	1	1	0	0	0	0
C	0	0	0	1	0	1	1	1

Πίνακας 5: Boolean πίνακας ωφέλειας αλλά με "threshold" 3

Από εκεί και πέρα, υπολογίζουμε τις αποστάσεις Jaccard όπως στο ερώτημα (a). Τα αντίστοιχα αποτελέσματα φαίνονται στον πίνακα 6.

	B	C
A	$2/5$	$2/6 = 1/3$
B	-	1/6

Πίνακας 6: Jaccard distance μεταξύ ζευγών χρηστών

(d) Όμοια με (b). Υπολογίζουμε πρώτα τις ευκλείδιες νόρμες:

$$\|v_A\| = \sqrt{4} = 2$$
$$\|v_B\| = \sqrt{3}$$
$$\|v_C\| = \sqrt{4} = 2$$

και μετά τα εσωτερικά γινόμενα (= μεγέθη των τομών):

$$v_A \cdot v_B = 2$$
$$v_A \cdot v_C = 2$$
$$v_B \cdot v_C = 1$$

Οι cosine distances φαίνονται στον πίνακα 7.

	B	C
A	$1 - 2/(2\sqrt{3}) = 1 - \sqrt{3}/3$	$1 - 2/4 = 1/2$
B	-	$1 - 1/(2\sqrt{3}) = 1 - \sqrt{3}/6$

Πίνακας 7: Cosine distance μεταξύ ζευγών χρηστών

(e) Ξεκινάμε υπολογίζοντας τον μέσο όρο των τιμών του κάθε χρήστη:

$$\bar{A} = \frac{4+5+5+1+3+2}{6} = \frac{20}{6} = \frac{10}{3}$$
$$\bar{B} = \frac{3+4+3+1+2+1}{6} = \frac{14}{6} = \frac{7}{3}$$
$$\bar{C} = \frac{2+1+3+4+5+3}{6} = \frac{18}{6} = 3$$

Αφαιρώντας, τώρα, από κάθε μη κενό κελί τον μέσο όρο του αντίστοιχου χρήστη, ο πίνακας ωφέλειας γίνεται όπως φαίνεται στον 8.

	a	b	c	d	e	f	g	h
A	$2/3$	$5/3$		$5/3$	$-7/3$		$-1/3$	$-4/3$
B		$2/3$	$5/3$	$2/3$	$-4/3$	$-1/3$	$-4/3$	
C	-1		-2	0		1	2	0

Πίνακας 8: Normalized Utility Matrix

(f) Για να υπολογίσουμε τις cosine distances, θα θεωρήσουμε καταρχάς τα κενά κελιά ως 0. Από εκεί και πέρα, δεν αλλάζει κάτι σε σχέση με τον τύπο που χρησιμοποιήσαμε στο (b), εκτός φυσικά από το ότι εδώ δεν έχουμε 0-1 και άρα το εσωτερικό γινόμενο δεν ταυτίζεται με το μέγεθος της τομής.

Υπολογίζουμε πρώτα τις ευκλείδιες νόρμες:

$$\begin{aligned}
\|v_A\|_2 &= \frac{1}{3}\sqrt{2^2+5^2+5^2+(-7)^2+(-1)^2+(-4)^2} \\
&= \frac{1}{3}\sqrt{4+25+25+49+1+16} = \frac{1}{3}\sqrt{120} = \frac{1}{3}\sqrt{8\cdot 3\cdot 5} = \frac{2}{3}\sqrt{30} \\
\|v_B\|_2 &= \frac{1}{3}\sqrt{2^2+5^2+2^2+(-4)^2+(-1)^2+(-4)^2} \\
&= \frac{1}{3}\sqrt{4+25+4+16+1+16} = \frac{1}{3}\sqrt{66} \\
\|v_C\|_2 &= \sqrt{(-1)^2+(-2)^2+1^2+2^2} \\
&= \sqrt{1+4+1+4} = \sqrt{10}
\end{aligned}$$

και μετά τα εσωτερικά γινόμενα:

$$\begin{aligned}
v_A \cdot v_B &= (\frac{1}{3})^2(2\cdot 0 + 5\cdot 2 + 0\cdot 5 + 5\cdot 2 + (-7)\cdot(-4) + 0\cdot(-1) + (-1)\cdot(-4) + (-4)\cdot 0) \\
&= \frac{1}{9}(10 + 10 + 28 + 4) = \frac{52}{9} \\
v_A \cdot v_C &= \frac{1}{3}(2\cdot(-1) + 5\cdot 0 + 0\cdot(-2) + 5\cdot 0 + (-7)\cdot 0 + 0\cdot 1 + (-1)\cdot 2 + (-4)\cdot 0) \\
&= \frac{1}{3}(-2 - 2) = -\frac{4}{3} \\
v_B \cdot v_C &= \frac{1}{3}(0\cdot(-1) + 2\cdot 0 + 5\cdot(-2) + 2\cdot 0 + (-4)\cdot 0 + (-1)\cdot 1 + (-4)\cdot 2 + 0\cdot 0) \\
&= \frac{1}{3}(-10 - 1 - 8) = -\frac{19}{3}
\end{aligned}$$

Με βάση τα παραπάνω, μπορούμε να υπολογίσουμε τα συνημίτονα των γωνιών μεταξύ των 3 διανυσμάτων:

$$\begin{aligned}
\frac{v_A \cdot v_B}{\|v_A\|_2\|v_B\|_2} &= \frac{52/9}{\frac{2}{3}\sqrt{30}\frac{1}{3}\sqrt{66}} = \frac{52}{2\sqrt{2\cdot 3\cdot 5\cdot 2\cdot 3\cdot 11}} = \frac{26}{6\sqrt{55}} = \frac{13}{3\sqrt{55}} \\
\frac{v_A \cdot v_C}{\|v_A\|_2\|v_C\|_2} &= -\frac{4/3}{\frac{2}{3}\sqrt{30}\sqrt{10}} = -\frac{2}{10\sqrt{3}} = -\frac{\sqrt{3}}{15} \\
\frac{v_B \cdot v_C}{\|v_B\|_2\|v_C\|_2} &= -\frac{19/3}{\frac{1}{3}\sqrt{66}\sqrt{10}} = -\frac{19}{2\sqrt{165}}
\end{aligned}$$

Τελικά, οι cosine distances φαίνονται συγκεντρωτικά στον πίνακα 7.

	B	C
A	$1 - \frac{13}{3\sqrt{55}} \approx 0.4157$	$1 + \frac{\sqrt{3}}{15} \approx 1.11547$
B	-	$1 + \frac{19}{2\sqrt{165}} \approx 1.73957$

Πίνακας 9: Cosine distance μεταξύ ζευγών χρηστών

MMDS, Άσκηση 9.3.2

Επαναλαμβάνουμε τον πίνακα ωφέλειας στο 10.

	a	b	c	d	e	f	g	h
A	4	5		5	1		3	2
B		3	4	3	1	2	1	
C	2		1	3		4	5	3

Πίνακας 10: Πίνακας ωφέλειας

(a) Κατά τις οδηγίες της εκφώνησης, αντικαθιστούμε 3, 4, 5 με 1 και 1, 2, κενό με 0 (11).

	a	b	c	d	e	f	g	h
A	1	1	0	1	0	0	1	0
B	0	1	1	1	0	0	0	0
C	0	0	0	1	0	1	1	1

Πίνακας 11: Πίνακας ωφέλειας με αντικαταστάσεις

Στην συνέχεια, για ευκολία, ξεκινάμε υπολογίζοντας εκ των προτέρων την απόσταση Jaccard μεταξύ των στηλών (χρησιμοποιώντας, φυσικά, τον ορισμό όπως τον είδαμε και στην προηγούμενη άσκηση).

Ένα επιπλέον κίνητρο για να το κάνουμε αυτό είναι ότι κατά το clustering η εκφώνηση μας λέει να θεωρήσουμε ως απόσταση δύο clusters την ελάχιστη απόσταση μεταξύ στοιχείων τους. Αυτό σημαίνει ότι όλες οι αποστάσεις που πρόκειται να συναντήσουμε σίγουρα θα είναι, τελικά, αποστάσεις μεταξύ ζευγών σημείων του "dataset" μας, δηλαδή των στηλών του πίνακα ωφέλειας (σε αντίθεση με περιπτώσεις όπως τα centroids, όπου εκεί δημιουργούμε σημεία που δεν υπάρχουν στα αρχικά μας, και άρα μπορεί να χρειαστεί να υπολογίσουμε και νέες αποστάσεις).

Ο πίνακας των αποστάσεων Jaccard όλων των items (στηλών) ανά δύο φαίνεται στο 12.

	b	c	d	e	f	g	h
a	1/2	1	2/3	1	1	1/2	1
b	-	1/2	1/3	1	1	2/3	1
c	-	-	2/3	1	1	1	1
d	-	-	-	1	2/3	1/3	2/3
e	-	-	-	-	1	1	1
f	-	-	-	-	-	1/2	0
g	-	-	-	-	-	-	1/2

Πίνακας 12: Jaccard distance μεταξύ ζευγών αντικειμένων

Ξεκινάμε, λοιπόν, τώρα, το ιεραρχικό clustering κατά τα γνωστά:

1. Κάθε item, ως μονοσύνολο, αποτελεί ένα ξεχωριστό cluster.

2. Η ελάχιστη απόσταση είναι το 0, μεταξύ των αντικειμένων f, h, οπότε αυτά τα δύο θα κάνουν την πρώτη συγχώνευση και θα γίνουν ένα cluster $\{f, h\}$.

3. Η δεύτερη μικρότερη απόσταση (αφού το 0 είναι πλέον μόνο intracluster και δεν "παίζει") είναι το $1/3$. Αυτή εμφανίζεται μεταξύ των b, d και d, g. Αυθαίρετα επιλέγουμε να συγχωνεύσουμε πρώτα τα b, d σε ένα νέο cluster 2 αντικειμένων, το $\{b, d\}$.

4. Παρομοίως, η επόμενη συγχώνευση θα είναι μεταξύ του cluster $\{b, d\}$ και του μεμονωμένου σημείου g, αφού η ελάχιστη απόσταση είναι $1/3$, σύμφωνα με τα όσα είπαμε πριν. Άρα, σχηματίζεται το cluster $\{b, d, g\}$.

5. Έχοντας τελειώσει και με την απόσταση $1/3$, η επόμενη μικρότερη απόσταση είναι το $1/2$, το οποίο βέβαια βλέπουμε ότι είναι η απόσταση πολλών ζευγών. Αυθαίρετα επιλέγουμε το πρώτο που βλέπουμε στον πίνακα, δηλαδή το a, b. Έτσι, το προηγούμενο cluster μεγαλώνει ακόμα περισσότερο και γίνεται $\{a, b, d, g\}$.

Σε αυτό το σημείο σταματάμε, καθώς έχουν σχηματιστεί 4 clusters. Αυτά είναι τα εξής:

- $\{a, b, d, g\}$
- $\{f, h\}$
- $\{c\}$
- $\{e\}$

(b) Δεδομένων των παραπάνω clusters, υπολογίζουμε τον πίνακα ωφέλειας για τα clusters των αντικειμένων, παίρνοντας για κάθε γραμμή το μέσο όρο των μη κενών τιμών των στηλών του cluster. Από την εκφώνηση καταλαβαίνουμε ότι ζητείται να κάνουμε τους υπολογισμούς εδώ με βάση τον αρχικό πίνακα ωφέλειας, και όχι τον boolean.

Ο τελικός πίνακας παρουσιάζεται στο 13.

	$\{a, b, d, g\}$	$\{f, h\}$	$\{c\}$	$\{e\}$
A	17/4	2		1
B	7/3	2	4	1
C	10/3	7/2	1	

Πίνακας 13: Πίνακας ωφέλειας για τα item clusters της (a)

(c) Ακολουθούμε τη διαδικασία κανονικά, όπως και στην προηγούμενη άσκηση: υπολογίζουμε τα 3 εσωτερικά γινόμενα και τις 3 ευκλείδιες νόρμες για τους χρήστες (δηλαδή τις γραμμές του πίνακα 13), και υπολογίζουμε το τελικό πηλίκο μέσω του τύπου του συνημιτόνου.

Εσωτερικά γινόμενα:

$$
\begin{aligned}
v_A \cdot v_B &= \frac{17}{4}\frac{7}{3} + 4 + 0 + 1 = \frac{119}{12} + 5 \\
v_A \cdot v_C &= \frac{17}{4}\frac{10}{3} + 2\frac{7}{2} + 0 + 0 = \frac{85}{6} + 7 \\
v_B \cdot v_C &= \frac{7}{3}\frac{10}{3} + 2\frac{7}{2} + 4 + 0 = \frac{70}{9} + 7 + 4 = \frac{70}{9} + 11
\end{aligned}
$$

Ευκλείδιες νόρμες:

$$
\begin{aligned}
\|v_A\|_2 &= \frac{1}{4}\sqrt{17^2 + 4^2 \cdot 2^2 + 0 + 4^2 \cdot 1^2} = \frac{1}{4}\sqrt{369} = \frac{1}{4}\sqrt{9 \cdot 41} = \frac{3}{4}\sqrt{41} \\
\|v_B\|_2 &= \frac{1}{3}\sqrt{7^2 + 3^2 \cdot 2^2 + 3^2 \cdot 4^2 + 3^2 \cdot 1^2} = \frac{1}{3}\sqrt{238} \\
\|v_C\|_2 &= \frac{1}{6}\sqrt{100 \cdot 2^2 + 49 \cdot 3^2 + 1 \cdot 6^2 + 0} = \frac{1}{6}\sqrt{877}
\end{aligned}
$$

Cosine similarities:

$$\cos(v_A, v_B) = \frac{\frac{119}{12} + 5}{\frac{3}{4}\sqrt{41}\frac{1}{3}\sqrt{238}} = \frac{119 + 60}{3\sqrt{9758}} = \frac{179}{3\sqrt{9758}} \approx 0.604$$

$$\cos(v_A, v_C) = \frac{\frac{85}{6} + 7}{\frac{3}{4}\sqrt{41}\frac{1}{6}\sqrt{877}} = \frac{4(85 + 42)}{3\sqrt{35957}} = \frac{508}{3\sqrt{35957}} \approx 0.893$$

$$\cos(v_B, v_C) = \frac{\frac{70}{9} + 11}{\frac{1}{3}\sqrt{238}\frac{1}{6}\sqrt{877}} = \frac{2(70 + 99)}{\sqrt{208726}} = \frac{338}{\sqrt{208726}} \approx 0.74$$