ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικόν και Καποδιστριακόν Πανεπιστήμιον Αθηνών
ΙΔΡΥΘΕΝ ΤΟ 1837

Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Τεχνητή Νοημοσύνη Ι

Πρώτη Εργασία
Θεωρητικές Ασκήσεις

Συγγραφή: Χρήστος Νίκου
ΑΜ: 1115201800330

20 Οκτωβρίου 2022

Πρόβλημα 2

Απάντηση: Εφ' όσον ο κόμβος στόχου βρίσκεται σε βάθος $g \leq d$ ο αλγόριθμος πρώτα σε βάθος με επαναληπτική εκβάθυνση θα πραγματοποιήσει ακριβώς g φορές τον αλγόριθμο αναζήτησης κατά βάθος, μια φορά για κάθε έναν απ' τους υπογράφους (ή υποδένδρα) που αποτελούνται από τα πρώτα $1, 2, \ldots, g$ επίπεδα του αρχικού γράφου. Τώρα, αφού ο παράγοντας διακλάδωσης είναι ίσος με b και ο κόμβος στόχου βρίσκεται σε βάθος $g \leq d$, ο αλγόριθμος θα δημιουργήσει σίγουρα gb κόμβους που αντιστοιχούν στο 1ο επίπεδο, $(g-1)b^2$ κόμβους που αντιστοιχούν στο 2ο επίπεδο, $(g-2)b^3$ στο 3ο επίπεδο κ.ο.κ. Συνεπώς, φτάνοντας μέχρι το $(g-1)$-οστό επίπεδο ο αλγόριθμος και στη χείριστη και στη καλύτερη περίπτωση θα έχει δημιουργήσει σίγουρα $gb + (g-1)b^2 + \cdots + 2b^{g-1}$ κόμβους.

(i) **Μικρότερος αριθμός κόμβων.** Τώρα, στην καλύτερη περίπτωση ο κόμβος στόχου που βρίσκεται στο επίπεδο g θα βρίσκεται στην ¨αριστερή¨ μεριά του δέντρου, δηλαδή, θα είναι ο πρώτος κόμβος που θα εξετάσει ο αλγόριθμος απ' το g-οστό επίπεδο. Έτσι, ο αλγόριθμος κατά τη g-οστή επανάληψη, θα δημιουργήσει ένα μόνο μονοπάτι το οποίο θα καταλήγει στον στόχο. Δηλαδή, θα παραγάγει επιπλέον gb κόμβους πέρα από τους $gb + (g-1)b^2 + \cdots + 2b^{g-1}$ κόμβους που θα έχει δημιουργήσει από τις πρώτες $g-1$ επαναλήψεις. Άρα, ο αλγόριθμος θα δημιουργήσει συνολικά

$$gb + (g-1)b^2 + \cdots + 2b^{g-1} + gb$$

κόμβους στην καλύτερη περίπτωση.

(ii) **Μεγαλύτερος αριθμός κόμβων.** Τώρα, στη χειρότερη περίπτωση ο κόμβος στόχου θα βρίσκεται στο δεξιότερο μέρος του δένδρου. Σε αυτή την περίπτωση περίπτωση, πέρα από τους $gb+(g-1)b^2+\cdots+2b^{g-1}$ κόμβους ο αλγόριθμος θα παραγάγει ακόμα όλους τους κόμβους του g-οστού επιπέδου, δηλαδή ακόμα b^g κόμβους. Συνεπώς, στη χειρότερη περίπτωση ο αλγόριθμος θα παραγάγει

$$gb + (g-1)b^2 + \cdots + 2b^{g-1} + b^g$$

το πλήθος κόμβους.

Πρόβλημα 3

Απάντηση: (α) Ισχυριζόμαστε ότι η ευρετική συνάρτηση h του προβλήματος είναι *παραδεκτή (admissible)* και *συνεπής (consistent)*.

Συνέπεια: Πρώτα δείχνουμε ότι η h είναι συνεπής. Για να το κάνουμε αυτό πρέπει να δείξουμε ότι για κάθε κόμβο n του προβλήματος και για κάθε διάδοχο n' του n που παράγεται από οποιαδήποτε ενέργεια α ισχύει ότι

$$h(n) \leq c(n, \alpha, n') + h(n'), \tag{3.1}$$

όπου $c(n, \alpha, n')$ είναι το κόστος της μετάβασης από τον κόμβο n στο n' μέσω της ενέργειας α. Για να δείξουμε ότι η h είναι συνεπής, εξετάζουμε ότι η ικανοποιείται η (3.1) ελέγχοντας έναν-έναν τους κόμβους/καταστάσεις του προβλήματος που έχουν τουλάχιστον έναν διάδοχο. Στον Πίνακα 1 βλέπουμε αυτούς τους κόμβους μαζί με τους διαδόχους τους και τις αντίστοιχες τιμές που εμπλέκονται στην (3.1).

Πίνακας 1: Ο έλεγχος της συνέπειας για την ευρετική συνάρτηση h

Κόμβος	Διάδοχοι	$c(n, \alpha, n') + h(n')$	$h(n)$	Ισχύς της (3.1)
ts	mail	32	23	ΝΑΙ
b3	b1, b4	17, 18	17	ΝΑΙ
b1	c2, b2	13, 21	13	ΝΑΙ
c2	c1, c3	10, 18	10	ΝΑΙ
c1	c3	20	6	ΝΑΙ
b2	b4	21	15	ΝΑΙ
b4	o109	32	24	ΝΑΙ
o109	o111, o119	31, 37	24	ΝΑΙ
o119	o123, storage	13, 19	11	ΝΑΙ
o123	r123, o125	4, 11	4	ΝΑΙ

Όπως βλέπουμε και απ' τον παραπάνω πίνακα η (3.1) ισχύει για όλους τους κόμβους που έχουν τουλάχιστον έναν διάδοχο πράγμα το οποίο αποδεικνύει ότι η h είναι συνεπής.

Η h είναι παραδεκτή: Για να δείξουμε ότι η h είναι παραδεκτή πρέπει να δείξουμε ότι η h δεν υπερεκτιμά το κόστος που έχουμε για να φτάσουμε στον στόχο. Πιο συγκεκριμένα, για κάθε κόμβο n για τον οποίο υπάρχει μονοπάτι προς τον κόμβο στόχο, αν θεωρήσουμε $h^*(n)$ το κόστος της βέλτιστης διαδρομής από τον κόμβο n μέχρι τον στόχο τότε θα πρέπει να ισχύει

$$h(n) \leq h^*(n). \tag{3.2}$$

Για να αποδείξουμε ότι η h είναι παραδεκτή δείχνουμε ότι κάθε συνεπής είναι και παραδεκτή:

Κάθε συνεπής είναι και παραδεκτή. Έστω n μια κατάσταση και έστω $h^*(n)$ το κόστος της βέλτιστης διαδρομής από τον κόμβο n σε έναν κόμβο στόχου. Χωρίς βλάβη της γενικότητας, θεωρούμε ότι το βέλτιστο μονοπάτι περιγράφεται μέσω της $n \to n_1 \to \cdots \to n_k$. Δηλαδή, για να φτάσουμε στον στόχο n_k με βέλτιστο τρόπο από το κόμβο n πηγαίνουμε στον κόμβο n_1 ύστερα στον n_2 κ.ο.κ. Τότε, το κόστος $h^*(n)$ της βέλτιστης διαδρομής από τον κόμβο n θα περιγράφεται μέσω της

$$h^*(n) = c(n, \alpha, n_1) + \sum_{j=1}^{k-1} c(n_j, \alpha_j, n_{j+1}). \tag{3.3}$$

Τώρα, αφού η h είναι συνεπής θα έχουμε ότι

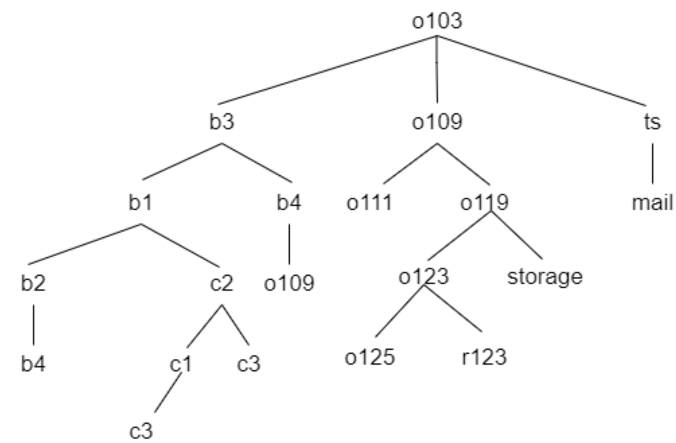
$$h(n) \leq c(n, \alpha, n_1) + h(n_1) \quad \text{και} \quad h(n_j) \leq c(n_j, \alpha_j, n_{j+1}) + h(n_{j+1}), \tag{3.4}$$

για κάθε $j = 1, \ldots, k-1$. Χρησιμοποιώντας την (3.4) διαδοχικά θα έχουμε

$$\begin{aligned} h(n) &\leq c(n, \alpha, n_1) + h(n_1) \leq c(n, \alpha, n_1) + c(n_1, \alpha_1, n_2) + h(n_2) \\ &\leq \cdots \leq \underbrace{c(n, \alpha, n_1) + \sum_{j=1}^{k-1} c(n_j, \alpha_j, n_{j+1})}_{h^*(n)} + \cancelto{0}{h_k(n)} = h^*(n), \end{aligned}$$

καταλήγοντας στο ότι $h(n) \leq h^*(n)$, το οποίο είναι και το ζητούμενο. $\square$

(β) Για να βρούμε με ποια σειρά βγαίνουν οι κόμβοι από τη λίστα «σύνορο» (fringe) σχεδιάζουμε πρώτα το δένδρο του προβλήματος. Έτσι, έχουμε την παρακάτω εικόνα.



Σχήμα 1: Το δένδρο του προβλήματος για το ρομπότ.

1. Αναζήτηση πρώτα σε πλάτος: Για να βρούμε τη σειρά με την οποία εξάγονται οι κόμβοι από το σύνορο στη περίπτωση της αναζήτησης πρώτα σε πλάτος διασχίζουμε το δένδρο του Σχήματος 1 όπως στον αλγόριθμο BFS. Έτσι, η σειρά με την οποία βγαίνουν οι κόμβοι σε αυτή την περίπτωση είναι η εξής: o103, b3, o109, ts, b1, b4, o111, o119, mail, b2, c2, o109, o123, storage, b4, c1, c3, o125, r123.

2. Αναζήτηση πρώτα σε βάθος: Εδώ οι κόμβοι εξάγονται με την ίδια σειρά που επισκεφτόμαστε τους κόμβους στο δένδρο κατά τον αλγόριθμο DFS. Έτσι, έχουμε την εξής σειρά: o103, b3, b1, b2, b4, c2, c1, c3, c3, b3, b4, o109, o109, o111, o119, o123, o125, r123.

3. Αναζήτηση πρώτα σε βάθος με επαναληπτική εκβάθυνση: Σε αυτή την περίπτωση εφαρμόζουμε διαδοχικά στα επίπεδα 1,2,3, 4, 5 τον αλγόριθμο DFS μέχρις ότου να φτάσουμε στον κόμβο στόχου r123. Έτσι η σειρά σε αυτή την περίπτωση δίνεται μέσω της ακολουθίας: o103, b3, o109, ts, o103, be, b1, b4, o109, o111, o119, ts, mail, o103, b3, b1, b2, c2 , b4, o109, o109, o111, o119, o123, storage, ts, mail, o103, b3, b1, b2, b4, c2, c1, c3, b4, o109, o109, o111, o119, o123, o125, r123.

4. Άπληστη αναζήτηση πρώτα στον καλύτερο με ευρετική συνάρτηση την h: Σε αυτή την περίπτωση διαλέγεται ο κόμβος με τη μικρότερη τιμή στη συνάρτηση h για να εξαχθεί. Έτσι έχουμε την εξής σειρά: o103, b3, b1, b1, c2, c3, b2, b4, o109, o119, o123, r123.

5. A^* με ευρετική συνάρτηση την h: Σε αυτή την περίπτωση επιλέγεται ο κόμβος του συνόρου με τη μικρότερη τιμή στην ποσότητα $f(n) + h(n)$. Η σειρά που προκύπτει με αυτή την προσέγγιση είναι η εξής: o103, b3, b1, c2, c1, b4, c3, ts, o109, o119, o123, r123.

Πρόβλημα 4

(α) Για να ορίσουμε ένα πρόβλημα αναζήτησης με μαθηματικό τρόπο θα πρέπει να προσδιορίσουμε τα εξής 4 στοιχεία:

1. Αρχική κατάσταση
2. Συνάρτηση διαδόχων: ζευγάρια της μορφής $\langle$ενέργεια, διάδοχος$\rangle$.
3. Έλεγχος στόχου και
4. Κόστος διαδρομής.

Για το πρόβλημα μεταφοράς των πακέτων από το ρομπότ στα διάφορα δωμάτια έχουμε τα εξής:

- Ως **αρχική κατάσταση** θεωρούμε ένα ζευγάρι της μορφής $\langle$δωμάτιο, κατάσταση πακέτων$\rangle$, όπου η πρώτη συνιστώσα υποδηλώνει το δωμάτιο στο οποίο βρίσκεται το ρομπότ και η δεύτερη συνιστώσα περιέχει ένα *λεξικό (dictionary)* με *κλειδιά (keys)* τα πακέτα προς παράδοση και *τιμές (values)* τις ενδείξεις «1» σε περίπτωση που έχει παραδοθεί το πακέτο και «0» σε αντίθετη περίπτωση. Έτσι, με βάση τα παραπάνω η αρχική κατάσταση περιγράφεται με απ' το ζευγάρι που η πρώτη συνιστώσα είναι το δωμάτιο «mail» και η κατάσταση των πακέτων έχουν όλα την ένδειξη «0», πράγμα το οποίο σημαίνει ότι δεν έχει παραδοθεί κανένα πακέτο στον προορισμό του.
- Τώρα με βάση τους παραπάνω ορισμούς η **κατάσταση στόχου** είναι ένα ζευγάρι της μορφής $\langle$δωμάτιο1, κατάσταση πακέτων$\rangle$ όπου η πρώτη συνιστώσα αντιστοιχεί σε ένα οποιοδήποτε δωμάτιο μπορεί να βρεθεί το ρομπότ και η δεύτερη στο λεξικό με τιμές την ένδειξη «1» για όλα τα κλειδιά, πράγμα το οποίο υποδηλώνει ότι το ρομπότ βρίσκεται στο δωμάτιο «δωμάτιο1» έχοντας παραδώσει όλα τα πακέτα. Να σημειώσουμε εδώ ότι δε χρειάζεται να απαιτήσουμε απ' το ρομπότ να βρίσκεται στην

τελική κατάσταση στο δωμάτιο «mail» γιατί αν το ρομπότ βρεθεί σε ένα οποιοδήποτε δωμάτιο έχοντας παραδώσει όλα τα πακέτα τότε το μόνο που έχουμε να κάνουμε για λάβουμε υπ'όψιν την επιστροφή του ρομπότ στο δωμάτιο «mail» είναι να προσθέσουμε στο κόστος διαδρομής την απόσταση επιστροφής στο δωμάτιο «mail» (βλέπε και παρακάτω στον ορισμό του κόστους διαδρομής).

- Το ρομπότ μπορεί να εκτελεί δύο **ενέργειες:** 1) Μετακίνηση από το δωμάτιο1 στο δωμάτιο2 για παράδοση πακέτου (παράδοση) και 2) μετακίνηση από το δωμάτιο1 στο δωμάτιο2 για παραλαβή πακέτου (παραλαβή). Έτσι, η **συνάρτηση διαδόχων** περιγράφεται μέσω της αντιστοίχισης

$$\langle \text{δωμάτιο1, ενέργεια, δωμάτιο2} \rangle \mapsto \langle \text{δωμάτιο2, κατάσταση πακέτων} \rangle,$$

  η οποία ερμηνεύεται ως «μετακίνηση από το δωμάτιο1 στο δωμάτιο2 μέσω της ενέργειας ¨ενέργεια¨», όπου ενέργεια $\in \{\text{παράδοση}, \text{παραλαβή}\}$.

- Τέλος, για το **κόστος διαδρομής** θεωρούμε την απόσταση από την αρχική κατάσταση μέχρι οποιαδήποτε κατάσταση στόχου όπου έχουν παραδοθεί όλα τα πακέτα συν την απόσταση απ' το τελευταίο δωμάτιο που βρίσκεται το ρομπότ μέχρι το δωμάτιο «mail».

(β) Μια **παραδεκτή ευρετική συνάρτηση** για το πρόβλημα είναι η εξής: Για μια κατάσταση $\underbrace{\langle \text{δωμάτιο1, κατάσταση πακέτων} \rangle}_{n}$ μπορούμε να ορίσουμε

$$\begin{aligned} h(n) = {} & \text{«Η απόσταση από το κόμβο } n \text{ μέχρι τον κόμβο } n' \text{ όπου έχουν παραδοθεί όλα τα πακέτα} \\ & + \text{την απόσταση που απαιτείται για να μεταφερθεί το ρομπότ από το δωμάτιο του κόμβου } n' \\ & \quad \text{μέχρι το δωμάτιο «mail»».} \end{aligned}$$

Πιο αναλυτικά, στη περίπτωση που στην κατάσταση n υπάρχουν k υπολοιπόμενα πακέτα προς παράδοση, τότε αυτά περιγράφονται από ζευγάρια της μορφής $\langle \text{δωμάτιο}1k, \text{ δωμάτιο}2k \rangle$, όπου για το πακέτο $1 \leq i \leq k$ πρέπει το ρομπότ να το παραλάβει απ' το δωμάτιο$1k$ και να το παραδώσει στο δωμάτιο$2k$. Με αυτό τον συμβολισμό έχουμε ότι

$$h(n) = \sum_{j=1}^{k} \text{απόσταση} \left(\text{δωμάτιο}1j, \text{ δωμάτιο}2j \right) + \text{απόσταση} \left(\text{δωμάτιο}2k, \text{mail} \right), \tag{4.1}$$

όπου με τον συμβολίσμό απόσταση $(\text{δωμάτιο}1j, \text{ δωμάτιο}2j)$ εννοούμε την απόσταση μεταξύ των δωματίων δωμάτιο$1j$ και δωμάτιο$2j$. Η συνάρτηση h είναι παραδεκτή αφού για κάθε κατάσταση n που βρίσκεται το ρομπότ σίγουρα θα διανύσει αυτές τις αποστάσεις για να παραδώσει όλα τα υπολοιπόμενα πακέτα και αυτό συμβαίνει διότι το ρομπότ μπορεί να μεταφέρει μόνο ένα πακέτο τη φορά.

Πρόβλημα 5

Όπως γνωρίζουμε η αμφίδρομη αναζήτηση στη περίπτωση που και οι δύο αναζητήσεις χρησιμοποιούν BFS είναι πλήρης και βέλτιστη όταν ο παράγοντας διακλάδωσης b είναι πεπερασμένος. Τώρα, έχουμε τα εξής:

(α) **Αναζήτηση πρώτα σε πλάτος και αναζήτηση περιορισμένου βάθους:** Τώρα στην περίπτωση που η προς τα πίσω αναζήτηση γίνεται με αναζήτηση περιορισμένου βάθους τότε ο αλγόριθμος αμφίδρομης αναζήτησης **είναι πλήρης**, όταν ο παράγοντας διακλάδωσης είναι πεπερασμένος αλλά εν γένει **όχι βέλτιστος.** Το ότι είναι πλήρης όταν ο παράγοντας διακλάδωσης είναι πεπερασμένος προκύπτει απ' το γεγονός ότι η αναζήτηση κατα πλάτος πάντα καταλήγει σε μια λύση. Έτσι, ακόμα και αν δεν συναντηθούν οι δύο αναζητήσεις σε κάποιο σημείο ο προς τα εμπρός αλγόριθμος αναζήτησης που γίνεται πρώτα κατά πλάτος κάποια στιγμή θα βρει τον στόχο και η αμφίδρομη αναζήτηση θα τερματίσει επιστρέφοντας μια λύση. Παρ'όλα αυτά, ο αλγόριθμος δεν είναι βέλτιστος εφόσον στη περίπτωση που συναντηθούν τα μονοπάτια η προς τα πίσω αναζήτηση μπορεί να ακολουθήσει ένα μη βέλτιστο μονοπάτι προς τη λύση και τότε το τελικό μονοπάτι που θα προκύψει θα είναι μη βέλτιστο. Ακόμα και αν δεν συναντηθούν τα μονοπάτια, μόνο η προς τα εμπρός αναζήτηση μας εγγυάται ότι θα τερματίσει ο αλγόριθμος. Όμως, γνωρίζουμε ότι η αναζήτηση πρώτα κατά πλάτος δεν επιστρέφει πάντα (εκτός αν είναι ομοιόμορφα τα κόστη) μια βέλτιστη λύση.

(β) **Αναζήτηση με επαναληπτική εκβάθυνση και αναζήτηση περιορισμένου βάθους:** Σε αυτή την περίπτωση η αναζήτηση πάλι **είναι πλήρης** (όταν το b πεπερασμένο) αλλά **όχι βέλτιση.** Το ότι είναι πλήρης προκύπτει απ' το γεγονός ότι η αναζήτηση με επαναληπτική εκβάθυνση επιστρέφει πάντα μια λύση, και έτσι, ακόμα και αν δεν συναντηθούν οι δύο αναζητήσεις η προς τα εμπρός αναζήτηση θα φτάσει κάποια στιγμή στον στόχο. Ο λόγος που ο αλγόριθμος δεν είναι βέλτιστος είναι ο ίδιος όπως και στην περίπτωση (α).

(γ) A^* **και αναζήτηση περιορισμένου βάθους:** Σε αυτή την περίπτωση η αμφίδρομη αναζήτηση **είναι πλήρης αλλά όχι βέλτιστη.** Στη περίπτωση που ευρετική συνάρτηση είναι παραδεκτή, η ακόμα καλύτερα συνεπής, τότε η προς τα εμπρός αναζήτηση που γίνεται με A^* θα τερματίσει κάποια στιγμή έχοντας βρει μια λύση. Τώρα, για να δούμε ότι δεν είναι βέλτιστος, αυτό προκύπτει απ' το γεγονός ότι η αναζήτηση περιορισμένου βάθους μπορεί να δημιουργήσει ένα μη βέλτιστο μονοπάτι προς τη λύση και αυτό να συναντηθεί με κάποιο κόμβο της εμπρόσθιας αναζήτησης δίνοντας συνολικά ένα μη βέλτιστο μονοπάτι για τη λύση.

(δ) A^* **και** A^***:** Σε αυτή την περίπτωση, αν τουλάχιστον μια απ' τις δύο ευρετικές συναρτήσεις που χρησιμοποιούν οι επιμέρους αναζητήσεις είναι παραδεκτή ή συνεπής και ισχύει ότι $b < \infty$, τότε η αμφίδρομη αναζήτηση θα βρει μια λύση, και έτσι είναι πλήρης σε αυτή την περίπτωση. Τώρα, αν μια τουλάχιστον απ' τις δύο συναρτήσεις δεν είναι παραδεκτή, τότε ο αλγόριθμος μπορεί να βρει τη βέλτιστη λύση. Αν και οι δύο είναι παραδεκτές ή συνεπείς, τότε ο αλγόριθμος θα βρει τη βέλτιστη λύση.

Στον παρακάτω πίνακα συνοψίζουμε όλα τα παραπάνω.

Πίνακας 2: Ο έλεγχος της συνέπειας για την ευρετική συνάρτηση h

Κριτήριο	Περίπτωση (α)	Περίπτωση (β)	Περίπτωση (γ)	Περίπτωση (δ)
Πλήρης;	Ναι$^{\alpha}$	Ναι$^{\alpha}$	Ναι$^{\alpha,\beta}$	Ναι$^{\alpha,\gamma}$
Βέλτιστη;	Όχι	Όχι	Όχι	Ναι$^{\alpha,\delta}$

Οι επιφυλάξεις που εμφανίζονται στον πίνακα είναι οι εξής: $^{\alpha}$ ο παράγοντας διακλάδωσης είναι πεπερασμένος, $^{\beta}$ Η ευρετική συνάρτηση είναι παραδεκτή, $^{\gamma}$ τουλάχιστον μια απ' τις δύο συναρτήσεις είναι παραδεκτή, $^{\epsilon}$ και οι δύο ευρετικές συναρτήσεις είναι παραδεκτές.

Τέλος, όσον αφορά την αποδοτικότητα του ελέγχου για το αν οι δύο αναζητήσεις συναντιούνται σε κάποιον κόμβο αυτό μπορεί να γίνει με τη χρήση ενός πίνακα κατακερματισμού για το σύνορο μιας εκ των δύο αναζητήσεων. Για παράδειγμα, στην Python αυτή η δομή μπορεί να αναπαρασταθεί από ένα σύνολο (set) όπου ο έλεγχος αν υπάρχει ένα στοιχείο στο σύνολο ή όχι γίνεται με $O(1)$.