ΑΛΓΟΡΙΘΜΟΙ & ΠΟΛΥΠΛΟΚΟΤΗΤΑ

3η Σειρά γραπτών ασκήσεων

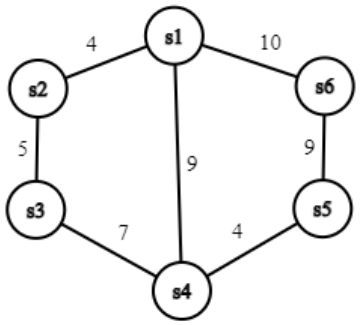
Ακαδημαϊκό έτος 2021-2022

7ο εξάμηνο

Νικόλας Μπέλλος | ΑΜ : el18183

Άσκηση 1 (MST με περιορισμούς)

α) Θα αποδείξουμε ότι το συγκεκριμένο άπληστο κριτήριο δεν οδηγεί στη βέλτιστη λύση αντιπαραθέτοντας ένα αντιπαράδειγμα.



Στο συγκεκριμένο γράφημα παρατηρούμε ότι για τη κορυφή **s1** αν το k είναι ίσο με 2, τότε σύμφωνα με το άπληστο κριτήριο θα επιλεγόντουσαν οι ακμές του s1 με βάρη 4 και 9 αντίστοιχα. Όμως τότε το MST που θα δημιουργούνταν, θα είχε συνολικό βάρος ίσο με 31 (4+5+9+4+9). Αυτό όμως δεν είναι η βέλτιστη λύση, καθώς αντί για την ακμή βάρους 9 θα έπρεπε να έχει επιλεγεί αυτή βάρους 10 η οποία οδηγεί στη δημιουργία ενός MST με συνολικό βάρος 30.

β)

Μία πρώτη ιδέα

Αν είχαμε υπολογίσει μία φορά το MST του G τότε θα γνωρίζαμε πόσες από τις ακμές του S ανήκουν σε αυτό. Αν είναι λιγότερες από K τότε θα πηγαίναμε και θα υπολογίζαμε για μία μία τις ακμές (u,v) που μένουν στον S, το μονοπάτι που ενώνει τις κορυφές u και v, δηλαδή τα άκρα της αντίστοιχης υποψήφιας ακμής. Από αυτά τα μονοπάτια θα κρατάγαμε το μεγαλύτερο βάρος ακμής που συναντήσαμε και θα το αφαιρούσαμε από το αντίστοιχο βάρος ακμής (u,v). Εν τέλει, θα συγκρίναμε αυτές τις διαφορές για όλες τις υποψήφιες ακμές και θα κρατάγαμε τη μικρότερη που θα μας υποδήλωνε το ζευγάρι ακμών που θα ανταλάσαμε. Τη διαδικασία αυτή θα την επαναλαμβάναμε μέχρι η S να έχει βαθμό ισο με K. Για τη περίπτωση τώρα που οι ακμές του S που ανήκουν στο MST ήταν περισσότερες από K, τότε θα ξανατρέχαμε τον αλγόριθμο του kruskal για τον υπολογισμό του MST, με τη παραλαγή όμως ότι αν ο βαθμός της κορυφής S ήταν ήδη K σε κάποιο βήμα, τότε οποιαδήποτε άλλη πιθανή ακμή που θα συνδεόταν με την S θα την απορρίπταμε από το MST.

Ωστόσο, η συγκεκριμένη λύση είναι αργή σε χρόνο εκτέλεσης και μπορούμε να βρούμε και κάτι καλύτερο.

Πολυπλοκότητα : O(ElogV)+O(A*V*K)

, όπου E : αριθμός ακμών, V : αριθμός κορυφών, A : Deg(S), k : ο βαθμός του S που θέλουμε να πετύχουμε

Μία καλύτερη ιδέα

Έστω ότι δημιουργούμε το MST ενός γραφήματος G και έστω η κορυφή S. Για τη κορυφή S έστω ότι έχει συνολικά 5 ακμές και στο MST περιέχονται μόνο οι 2 ακμές. Αν μία ακμή βρίσκεται στο MST τότε αυτό συμβαίνει γιατί είναι αρκετά μικρή σε σύγκριση με όλες τις εναλλακτικές που υπήρχαν εκείνη τη στιγμή στο priority queue. Χωρίς βλάβη της γενικότητας, όσο και αν μειώσουμε το βάρος της ακμής αυτής τότε εκείνη θα επιλεγόταν και πάλι για το MST. Επομένως αν μειώναμε το βάρος w και στις 5 ακμές του S τότε και πάλι θα υπήρχαν τουλάχιστον 2 ακμές στο MST. Αντίστοιχα, αν κάποια ακμή δεν ανήκει στο MST, όταν αυξήσουμε το βάρος της και τρέξουμε και πάλι έναν εκ των prim, kruskal, θα διαπιστώσουμε ότι και πάλι δεν θα ανήκει στο MST. Τι γίνεται όμως αν μία ακμή δεν ανήκε στο MST και μειώσουμε το βάρος της ? Τότε, αν η τιμή της ήταν αρκετά μικρή μία από τις φορές που είχε την ευκαιρία να επιλεγεί, θα επιλεγόταν στο MST και θα έπαιρνε τη θέση κάποιας άλλης ακμής, όπως το αντίθετο συμβαίνει και αν αυξήσουμε το βάρος μια κορυφής που ανήκε στο MST.

Ωστόσο, για να μην πάρει τη θέση μίας από τις ακμές της ίδιας κορυφής θα έπρεπε αυτή η αυξομείωση των βαρών να είναι αντίστοιχη για όλες τις ακμές του S ώστε η σχετική διαφορά των βαρών τους να μείνει σταθερή.

Θα κάνουμε λοιπόν binary search στην τιμή κατά την οποία θα πρέπει να αυξομειώσουμε όλες τις ακμές της κορυφής S. Αν για κάποια τιμή που δοκιμάζουμε μετά το τρέξιμο του αλγορίθμου MST ο βαθμός της S είναι μικρότερος από αυτόν που θα θέλαμε, τότε δοκιμάζουμε να μειώσουμε και άλλο τα βάρη των ακμών. Διαφορετικά, δοκιμάζουμε να τα αυξήσουμε.

Όσον αφορά την πολυπλοκότητα αυτού του αλγορίθμου, Το πλήθος των τιμών που θα δοκιμάσουμε θα είναι λογαριθμικό της μέγιστης τιμής των ακμών του γράφου και για κάθε τέτοια τιμή θα τρέξουμε ένα MST αλγόριθμο που κοστίζει O(ElogV).

Παρατηρήσεις :

Στη περίπτωση που ο κόμβος S έχει πολλαπλές ακμές με το ίδιο βάρος, τότε τρέχοντας prim ή kruskal θα πρέπει να υπάρχει κάποια προτεραιοποίηση των ακμών του S αν αυτοί κατά την ταξινόμηση (έχοντας αυξηθεί κατά K έχουν το ίδιο βάρος με κάποια άλλη ακμή του δέντρου, ώστε να υπάρχει κάποια συνέπεια στο τρόπο με τον οποίο γίνεται η ταξινόμηση και η σύγκριση στους αλγορίθμους του MST.

Τελική πολυπλοκότητα : $O(E*logV*logE_{max})$

Άσκηση 2 (Σχεδιασμός Videogame)

1)

Αρχική (αλλά όχι πολυωνυμικού χρόνου) ιδέα

Θα μπορούσαμε να ελέγξουμε όλα τα πιθανά μονοπάτια στο κατευθυνόμενο γράφο από τον s στον t, αλλά αυτό θα είχε πολυπλοκότητα της τάξεως του n! καθώς για κάθε κόμβο κάθε φορά θα είχαμε το πολύ άλλες n-1 επιλογές. Άμα σε τουλάχιστον ένα από αυτά τα μονοπάτια η ζωή δεν έπεφτε ποτέ κάτω από 0 τότε θα επιστρέφαμε ότι ο λαβύρινθος είναι r-ασφαλής.

Καλύτερη ιδέα

Το συγκεκριμένο πρόβλημα είναι μία παραλλαγή του προβλήματος bellman ford για εύρεση μεγαλύτερου μονοπατιού (μιας και έχουμε κατευθυνόμενο γράφο με αρνητικά βάρη). Στην υλοποίηση αυτή, σε κάθε κόμβο u θέλουμε να αποθηκεύουμε το μέγιστο άθροισμα μονοπατιού από τον S μέχρι εκείνο το κόμβο, λαμβάνοντας υπόψη στο άθροισμα μόνο ακμές οι οποίες αν τις προσθέσουμε δεν θα μας δώσουν αρνητικό άθροισμα σε κάποιο υπομονοπάτι από τον S στο u. Για αυτό, θα αρχικοποιήσουμε όλες τις ακμές σε -∞ εκτός από την s η οποία θα πάρει τη τιμή r και στο πρώτο iteration για όλες τις ακμές u → v θα κάνουμε τη παρακάτω σύγκριση : d(v) < d(u) + w(u,v). Αν η ανισότητα αληθεύει και η τιμή d(u) + w(u,v) είναι μεγαλύτερη του 0 τότε έχουμε βρεί έναν εναλλακτικό τρόπο για να φτάσουμε στο δωμάτιο v με περισσότερη ζωή από πριν. Αυτό τον αλγόριθμο θα τον επαναλάβουμε συνολικά n-1 φορές (όπως και στον bellman-ford) και άρα θα κάνουμε συνολικά N*M συγκρίσεις όπου η κάθε μία θα κοστίζει O(1). Επομένως, εφαρμόζοντας το παραπάνω αλγόριθμο, στο τέλος θα έχουμε τη μέγιστη τιμή ζωής που μπορούμε να έχουμε όταν βρεθούμε στο δωμάτιο t έχοντας ξεκινήσει με ζωή r από το δωμάτιο s. Αν αυτή είναι < 0 τότε ο λαβύριθμος δεν θα είναι r-ασφαλής γιατί αυτό σημαίνει ότι όλα τα μονοπάτια κάποια στιγμή μας οδήγησαν σε αρνητική ζωή. Αν πάλι η τιμή είναι ≥ 0 ο λαβήρινθος θα είναι r-ασφαλής. (Επειδή δεν μας ενδιαφέρει να βρούμε αρνητικό κύκλο μιας και ψάχνουμε το μέγιστο μονοπάτι δεν θα κάνουμε την τελευταία επανάληψη που γίνεται στο τέλος). Λόγω των επαναλήψεων πο περιγράψαμε παραπάνω, η πολυπλοκότητα θα είναι αντίστοιχη με αυτή του bellman-ford και επομένως θα ανάγεται σε :

Συνολική πολυπλοκότητα : O(E* V)

Απόδειξη ορθότητας

Για κάθε κόμβο αν μπορούμε να μεταβούμε σε αυτόν με αρκετή ζωή από πάνω από 1 μονοπάτι, θέλουμε η ζωή που θα μας απομένει να είναι η μέγιστη. Επομένως στη βάση του αυτός ο αλγόριθμος είναι ένας greedy αλγόριθμος. Ωστόσο υπάρχει και ο περιορισμός της ζωής, και για αυτό θεωρούμε ότι αν σε κάποιο κόμβο η ζωή μας πέσει κάτω από 0 αυτό θα είναι μονοπάτι μέγιστης αντίστασης και δεν θα λάβουμε υπόψη στο μονοπάτι (s→t) αυτήν την ακμή.

2)

Γενική ιδέα

Στη περίπτωση που υπάρχει κύκλος που ενισχύει τη δύναμη του παίκτη, τότε εκείνος θα μπορούσε να περιφέρεται στο κύκλο μέχρι να μαζέψει αρκετή ζωή ώστε να ξεπεράσει οποιοδήποτε τέρας. Σε αυτή τη περίπτωση επομένως, αρκεί να βρούμε αν μπορούμε αρχικά να φτάσουμε με αρκετή ζωή μέχρι κάποιο κόμβο του κύκλου και έπειτα αν ο κύκλος αυτός συνδέεται με τον κόμβο t. Όλα αυτά βέβαια, έχουν νόημα αν στην αρχική λύση μας (χωρίς κύκλους) η τιμή r δεν ήταν αρκετά μεγάλη για να φτάσουμε στην έξοδο. Αν λοιπόν μετά από (N-1)*M επαναλήψεις, η τιμή στην έξοδο είναι αρνητική, τότε θα ξανατρέξουμε τον αλγόριθμο του υποερωτήματος 1 για μία τελευταία φορά για όλες τις ακμές. Αν βρούμε ότι η τιμή κάποιας κορυφής πρέπει να αλλάξει, αυτό σημαίνει ότι υπάρχει κάποιος θετικός κύκλος στο γράφο, και επομένως θα μπορούμε να αυξήσουμε τη ζωή όσο θέλουμε και να βγούμε από το λαβύρινθο. Διαφορετικά, άμα καμία κορυφή δεν αλλάξει επιστρέφουμε ότι λαβύρινθος δεν είναι r-ασφαλής. Η πολυπλοκότητα θα είναι ίδια με πριν, καθώς προσθέσαμε μοναχά ακόμα μία επανάληψη για όλες τις ακμές.

Συνολική πολυπλοκότητα : O(E*V)

Απόδειξη ορθότητας

Η ορθότητα του αλγορίθμου προκύπτει από αυτή του bellman-ford για εύρεση μέγιστης διαδρομής s-t σε γράφο με θετικούς κύκλους

3)

Γενική ιδέα

Εκμεταλλευόμενοι τα προηγούμενα 2 ερωτήματα, για να βρούμε την ελάχιστη τιμή r που χρειάζεται για να βγούμε από τον λαβύρινθο, θα μπορούσαμε να δοκιμάζουμε διάφορες τιμές του r ώστε να δούμε αν ξεκινώντας με r ζωή μπορούμε να βγούμε ζωντανοί από το λαβύρινθο. Εκμεταλλευόμενοι το γεγονός ότι για μία τιμή του r και κάτω κανένας λαβύρινθος δεν είναι r-ασφαλής και αντίστοιχα από μία τιμή και πάνω όλοι οι λαβύρινθοι είναι r-ασφαλείς, για να βρούμε τη βέλτιστη τιμή, θα μπορούσαμε να κάνουμε binary search στη τιμή του r η οποία θα μπορεί να κυμαίνεται από 0 μέχρι και την απόλυτη τιμή του αρνητικού αθροίσματος των πόντων ζωής που χάνονται από τα τέρατα. Για κάθε λοιπόν τέτοια πιθανή τιμή του r, θα εφαρμόζαμε τον αλγόριθμο του ερωτήματος 2 (θεωρώντας ότι μπορεί να υπάρχουν κύκλοι θετικού μήκους) και άμα ο λαβύρινθος ήταν όντως r-ασφαλής τότε θα δοκιμάζαμε κάποια μικρότερη τιμή, διαφορετικά θα δοκιμάζαμε κάποια μεγαλύτερη, κάνοντας πάντα εκθετικά μεγάλα βήματα. Η **τελική πολυπλοκότητα** μίας τέτοιας υλοποίησης θα ήταν : **O(E*V* log(S))**, όπου S το άθροισμα των αρνητικών βαρών.

Εναλλακτική ιδέα

Αλλάζοντας την υλοποίηση των προηγούμενων δύο ερωτημάτων, θα μπορούσαμε να τρέξουμε μία άλλη παραλλαγή του αλγορίθμου bellman-ford στην οποία ο σκοπός θα είναι σε κάθε κόμβο u να μεγιστοποιήσουμε την ελάχιστη τιμή που θα συναντηθούμε σε κάποιο υπομονοπάτι της διαδρομής από τον S στο u. Για να το πετύχουμε αυτό, θα μπορούσαμε αυτή τη φορά θα θέσουμε την αρχική τιμή του

κόμβου S ίση με ο και στον αλγόριθμο να συμπεριλαμβάνουμε ακόμα και τις ακμές που αποφέρουν κάποιο αρνητικό μήκος υπομονοπατιού από τον s στον t. Για κάθε ακμή (u,v) θα είχαμε : m(v) = max(m(v), min(m(u), d(u)+w(u,v)), όπου m(v) το ελάχιστο μήκος υπομονοπατιού s-v' για το μονοπάτι s-v. Αυτή τη σχέση άμα τη τρέχαμε για όλες τις ακμές επί το πλήθος των κορυφών (΄όπως στο bellman-ford), θα μας έδινε τη τιμή m(t) η οποία θα μας έλεγε ότι σε μονοπάτι s-t, άμα ξεκινάγαμε με ονική ζωή, θα χάναμε το πολύ m(t) πόντους ζωής σε κάποιο σημείο. Αν αυτή η τιμή ήταν αρνητική πχ. -5, τότε η ελάχιστη τιμή του r θα ήταν 5 πόντοι ζωής. Διαφορετικά, αν ήταν θετική, η βέλτιστη τιμή του r θα ήταν το ο.
Η **πολυπλοκότητα** αυτής της υλοποίησης θα ήταν : **O(E*V)**, όπως και πριν, λόγω του ότι κάνουμε για V επαναλήψεις E συγκρίσεις στις ακμές.

Άσκηση 3 (Ταξίδι σε περίοδο ενεργειακής κρίσης)

1)

Για κάθε κόμβο i, θεωρώ ότι έχω την τιμή g(i), η οποία αντιπροσωπεύει πόση βενζίνη έχω στο ντεπόζιτο όταν βρεθώ σε εκείνον. Για τον εκάστοτε κόμβο i, θα ψάξω να βρώ τον κόμβο j, για τον οποίο να ισχύει $j > i$ και $c[j] < c[i]$ η $d_{ij} > B$. Στη περίπτωση που βρώ κάποιον κόμβο j για τον οποίο ισχύει $c[j] < c[i]$ και $d_{ij} \leq B$, Τότε θα γεμίσω το ντεπόζιτο μου τόσο όσο χρειάζεται ώστε να φτάσω σε εκείνη τη πόλη, δηλαδή θα αγοράσω d_{ij}-g(i) λίτρα βενζίνης, και θα συνεχίσω τον αλγόριθμο από εκείνον τον κόμβο j, μηδενίζοντας το ντεπόζιτο $g[j]=0$. Αν Δεν βρώ κάποιον κόμβο με μικρότερη τιμή σε ακτίνα B, τότε θα αγοράσω βενζίνη μέχρι να γεμίσει το ντεπόζιτο, δηλαδή θα αγοράσω B - g(i) λίτρα, και προχωρήσω στον επόμενο i+1 κόμβο. Στη χειρότερη περιπτωση όλες οι πόλεις θα έχουν γνησίως αύξουσα τιμή βενζίνης και άρα θα γίνουν n^2 επαναλήψεις άμα η χωρητικότητα του ντεποζίτου είναι πολύ μεγάλη. Άρα η πολυπλοκότητα αυτής της greedy λύσης θα είναι :

Τελική πολυπλοκότητα : $O(n^2)$

2)

Γενική ιδέα

Θα κάνω ένα BFS στο γράφο G και θα εφαρμόζω σταδιακά τον αλγόριθμο του πρώτου ερωτήματος. Για κάθε νέο κόμβο που θα συναντώ θα υπολογίζω το κόστος για να πάω από τον S στον κόμβο αυτό μέσω ενός συγκεκριμένου μονοπατιού που θα σχηματιστεί από τους πατεράδες των προηγούμενων κόμβων. Αν για ένα κόμβο v εχει υπολογιστεί προηγουμένως ένα κόστος διαδρομής s-v και τον ξανασυναντήσουμε μέσω μία ακμής u→v τότε θα υπολογίσουμε ξανά μία διαδρομή s-v με καινούργιο πατέρα του v τον u μόνο αν το κόστος που έχουμε υπολογίσει μέχρι στιγμής για να πάμε στον u είναι μικρότερο από που έχουμε υπολογίσει για να πάμε στον v. Έπειτα, θα ανανεώσουμε την τιμή του v με το μικρότερο κόστος. Στη χειρότερη περίπτωση, για κάθε ακμή του γράφου θα χρειαστεί να υπολογίσουμε μία καινούργια διαδρομή η οποία θα κοστίζει $O(n^2)$. Άρα η συνολική πολυπλοκότητα θα ανάγεται σε :

Τελική πολυπλοκότητα : $O(E*V^2)$

Άσκηση 4 (Επαναφορά της ομαλότητας στη χώρα των αλγορίθμων)

Γενική ιδέα - αλγόριθμος

Μπορούμε άμεσα να βγάλουμε δύο λογικά συμπεράσματα :

1. Για κάθε επίθεση έχει νόημα να επιτίθεται πάντα ο στρατός που είναι πιο κοντά κατά απόλυτη απόσταση
2. Για να καταστρέψουμε μία περιοχή ακτίνας d, θα πρέπει να συγκρίνουμε το κόστος για τη καταστροφή της βάσης και το άθροισμα για τα κόστη των επιμέρους επιθέσεων και να κρατήσουμε την "οικονομικότερη" λύση.

Το 2ο συμπέρασμα μας οδηγεί να ανάγουμε το συγκεκριμένο πρόβλημα σε αυτό της εύρεσης μέγιστης ροής (min-cut / max-flow). Η γενικότερη ιδέα είναι ότι αν πάμε να κάνουμε πολλές επιμέρους επιθέσεις σε εξτρεμιστές τότε το κόστος της επίθεσης στην αντίστοιχη βάση των εξτρεμιστών θα φράσει το συνολικό κόστος που χρειάζεται για να εξοντωθούν. Για αυτό το λόγο θα δημιουργήσουμε ένα κατευθυνόμενο γράφημα G με αρχικό και τελικό κόμβο s και t αντίστοιχα. Ενδιάμεσα θα υπάρχουν δύο επίπεδα κόμβων. Στο πρώτο επίπεδο θα συνδέσουμε τον s με όλους τους εξτρεμιστές και τα βάρη των ακμών θα είναι το κόστος επίθεσης του πιο κοντινού στρατού. Στο δεύτερο επίπεδο, θα συνδέσουμε όλους τους κόμβους του πρώτου επιπέδου με κόμβους που θα αναπαριστούν τις βάσεις των εξτρεμιστών και θα υπάρχει διάνυσμα από έναν εξτρεμιστή e_i σε κάποια βάση b_j μόνο αν έχουν απόσταση μικρότερη ή ίση από d. Το βάρος των ακμών αυτών θα είναι άπειρο. Τέλος, θα συνδέσουμε όλους τους κόμβους του 2ου επιπέδου (δηλαδή τις βάσεις) με τον τελικό κόμβο t και τα βάρη των ακμών αυτών θα είναι το ελάχιστο κόστος επίθεσης από το στρατό στην αντίστοιχη βάση αυτή. Ο τελικός γράφος θα είναι ένα DAG με συνολικό αριθμό κόμβων $2+N_e+N_b$. Με αυτή τη τεχνική έχουμε καταφέρει :

1. Μέσω των άπειρων βαρών μεταξύ εξτρεμιστών και βάσεων δίνεται η επιλογή να γίνεται επίθεση σε όποιο από τα 2 θέλουμε
2. Το πρώτο επίπεδο μας λέει ότι μπορούμε να δοκιμάσουμε να απολύσουμε επίθεση σε όλους τους επιμέρους εξτρεμιστές. Αν όμως κάποιο σύνολο επιθέσεων δεν είναι αρκετά "οικονομικό", τότε το κόστος επίθεσης στη αντίστοιχη βάση έρχεται σαν άνω φράγμα και μας εξασφαλίζει το ελάχιστο κόστος.

Για να φτιάξουμε το γράφημα πρέπει να ταξινομήσουμε τις θέσεις των ai, ei και bi στον οριζόντιο άξονα και αυτό θα μας πάρει χρόνο $O((N_a+N_e+N_b)\log(N_a+N_e+N_b))$.

Επιπλέον, για να λύσουμε το πρόβλημα, αυτό που μένει είναι να λύσουμε το πρόβλημα του max-flow, το οποίο μπορούμε εύκολα να το λύσουμε μέσω του αλγορίθμου ford-fulkerson. Ο αλγόριθμος αυτός έχει time complexity $O(|E|*e_{max})$ όπου στη συγκεκριμένη περίπτωση θα ισούται με $O((N_e+N_b+N_a*N_b)* X_{max})$, όπου Xmax η μεγαλύτερη μικρότερη απόσταση μεταξύ στρατού και εξτρεμιστών-βάσεων.

Άσκηση 5 (Αναγωγές και NP-πληρότητα)

1. Τακτοποίηση Ορθογωνίων Παραλληλογράμμων

Για το συγκεκριμένο πρόβλημα θα δοκιμάσουμε να κάνουμε αναγωγή από το NP-complete πρόβλημα του 2-Partition, στο οποίο πρέπει να χωρίσουμε ένα σετ ακεραίων σε δύο μέρη έτσι ώστε τα επιμέρους αθροίσματα των μερών αυτών να είναι ίδια.

Χωρίς να περιστρέψουμε τα επιμέρους παραλληλόγραμμα A_i, θα δοκιμάσουμε να δημιουργήσουμε ένα ορθογώνιο B, το οποίο θα έχει τέτοιο πλάτος ώστε να μπορούμε να βάλουμε μέσα του τα παραλληλόγραμμα και αυτά να μοιραστούν ακριβώς στα 2. Θα θεωρήσουμε ένα πλάτος ε του κάθε παραλληλογράμμου A_i, τέτοιο ώστε το εμβαδόν του πλέον να είναι ε x A_i και θα θεωρήσουμε και το διπλάσιο πλάτος για το ορθογώνιο B το οποίο θα έχει διάσταση 2ε x T, όπου $T = 1/2 * \Sigma a_i$. Αν επιλέξουμε κατάλληλα αυτόν τον αριθμό ε και τα ορθογώνια μπορούν να χωριστούν στα δύο μέρη του B, τότε απαντάμε ότι είναι δυνατόν να τοποθετηθούν. Διαφορετικά απαντάμε αρνητικά. Καθώς η αναγωγή μπορεί να γίνει σε πολυωνυμικό χρόνο και το πρόβλημα από το οποίο κάναμε την αναγωγή είναι NP-complete τότε και αυτό εδώ θα είναι NP-complete