

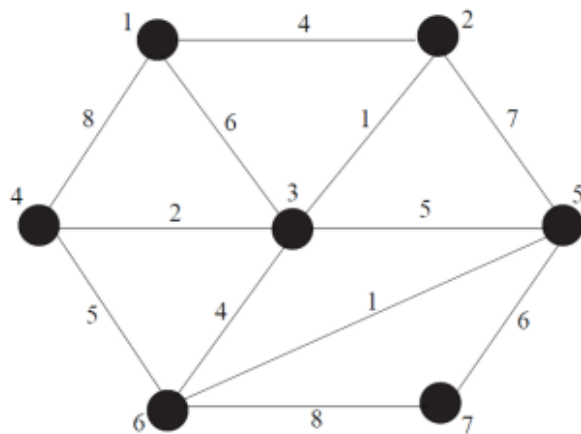
Συνδυαστική Βελτιστοποίηση

Εργαστηριακή αναφορά

Άσκηση 2

Σοφιανόπουλος Έκτορας 2017010016

Να γραφεί πρόγραμμα σε Matlab, Octave, C, C++, Python, Java... που να εφαρμόζει τον αλγόριθμο Dijkstra ή τον αλγόριθμο Ford (Belman-Ford) για την εύρεση της συντομότερης διαδρομής από τον κόμβο 1 στον τελικό κόμβο σε οποιοδήποτε γράφημα (για παράδειγμα στο ακόλουθο γράφημα, αρχικός κόμβος ο 1, τελικός ο 7. Επίσης, η φορά των τόξων, αν χρειάζεται, είναι από τον μικρότερο κόμβο σε αρίθμηση προς τον μεγαλύτερο κόμβο σε αρίθμηση). Στην αναφορά να περιγράφετε τον αλγόριθμο που αναπτύξατε αναλυτικά και να δίνεται η συντομότερη διαδρομή για το γράφημα που λύσατε.




Αρχικά κατασκευάστηκε η **συνάρτηση read_graph** η οποία δέχεται τις ακμές του γραφήματος ως “αρχικό node τελικό node weight” και τις αποθηκεύει σε ένα dictionary. Η εντολή defaultdict από τη βιβλιοθήκη collections χρησιμεύει καθώς μπορούμε να προχωρήσουμε στην τροποποίηση της χωρητικότητας μιας ακμής χωρίς να χρειαστεί να ελέγξουμε πρώτα αν είναι καταχωρημένη στο dictionary. Η συνάρτηση διαβάζει κάθε γραμμή του input2.txt εκτός από τις τελευταίες δύο και αποθηκεύει τα στοιχεία της ως : `graph[start][end]=int(weight)` δημιουργώντας έτσι το γράφημα. Έπειτα από τις δύο τελευταίες γραμμές δέχεται τον αρχικό και τελικό κόμβο από αυτούς του γραφήματος και τους αποθηκεύει. Επιπλέον η συνάρτηση επιστρέφει όλα τα nodes (`all_nodes`), συμπεριλαμβανομένου και του τελευταίου από το οποίο δεν ξεκινούν ακμές.

Η συνάρτηση dijkstras δέχεται ως ορίσματα το graph, την μεταβλητή nodes, καθώς και τον αρχικό και τον τελικό κόμβο. Αρχικοποιεί τις αποστάσεις για κάθε κόμβο σε άπειρο, εκτός από τον τελευταίο τον οποίο θέτει σε 0: “shortest_paths = {vertex: float(‘infinity’) for vortex in nodes}”, “shortest_paths[start] = 0”. Ύστερα, μέσω της βιβλιοθήκης heap που έχει γίνει import στο script βρίσκει τον επόμενο κόμβο με την ελάχιστη απόσταση και ενημερώνει τις αποστάσεις των γειτονικών κόμβων. Τελικά με μια δομή επανάληψης while ανακατασκευάζει τη συντομότερη διαδρομή από τον τελικό κόμβο στον αρχικό. Η συνάρτηση επιστρέφει τη συνολική ελάχιστη απόσταση από τον αρχικό στον τελικό κόμβο : shortest_paths[end] και την λίστα των κόμβων της διαδρομής που βρέθηκε αν αυτή πράγματι αρχίζει από τον αρχικό κόμβο.

Η if __name__ == “__main__” εκτελεί τον κώδικα εφόσον το script τρέχει ως κύριο πρόγραμμα. Πρώτα διαβάζει το γράφημα και τον αρχικό και τελικό κόμβο από το input2.txt και ύστερα καλεί την συνάρτηση. Τελικά ανοίγει το αρχείο εξόδου output2.txt και τυπώνει μέσα ότι επέστρεψε η συνάρτηση δηλαδή την συντομότερη διαδρομή και την τελική απόσταση, αφού ελέγξει πρώτα ότι ο αρχικός και ο τελικός κόμβος υπάρχουν στο σύνολο των κόμβων nodes και ότι έχει πράγματι βρεθεί ελάχιστη διαδρομή.

Το script τρέχει για οποιοδήποτε γράφημα επιθυμεί ο χρήστης, αρκεί να τροποποιήσει τα περιεχόμενα του αρχείου εισόδου.

 input2 - Σημειωματάριο

Αρχείο	Επεξεργασία	Μορφή	Προβολή	Βοήθεια
	1	2	4	
1	3	6		
1	4	8		
2	3	1		
2	5	7		
3	4	2		
3	5	5		
4	6	5		
5	6	1		
5	7	6		
6	3	4		
6	7	8		
1				
7				

Περιεχόμενα input2.txt