**Γιάννης Μπάρζας 2765**

**Ασκηση 1**

1. Αρχικοποιουμε τη λιστα **min\_node** με αρχικες τιμες τους ιδιους τους κομβους v,οπου για κάθε κομβο v θα αποθηκευει τον y=min (v).Μια λιστα **min\_weight** αρχικοποιημενη με το βαρος του κάθε κομβου,οπου για κάθε κομβο v θα αποθηκευει το βαρος w(y) του y=min (v) για τον κομβο v.Μια λιστα **saw** για τα bit σήμανσης παλι για κάθε κομβο οπου την αρχικοπουμε στην τιμη False.Τελος τη λιστα **nodes\_to\_compute** με ολους του κομβους του γραφηματος που προκειτε να υπολογιστουν.Τελος η λιστα **nodes\_done** αρχικοποιημενηκενη , που θα περιεχει σε κάθε βημα εκτελεσης τους κομβους που εχουν υπολογιστει σε αυτό το βημα ετσι ώστε να τους αφερεσουμε από την λιστα των κομβων που θα εκτελεστουν μελλοντικα (**nodes\_to\_compute)**.
2. Εκτελουμε τον αλγοριθμο υπολογισμού των ισχυρά συνδεδεμένων συνιστωσών μια φορα σε χρονο εκτελεσης Ο(m+n) που θα μας επιστρεψει τη λιστα των ισχυρά συνδεδεμένων συνιστωσών.
3. Στη συνεχεια για κάθε κομβο μεσα στη λιστα **nodes\_to\_compute**

μηδενιζουμε τα bit σήμανσης στη λιστα **saw** για όλους του κομβους βαζουμε την λιστα **nodes\_done** να είναι κενη και καλουμε **BFS** με αρχικο κομβο αυτόν που ειμαστε στη λιστα **nodes\_to\_compute**.Η επαναλυψη συνεχιζεται μεχρι να αδειασει η λιστα **nodes\_to\_compute**.

1. Η εκτελεσητου **BFS** ξεκιναει την διερευνηση από τον κομβο αφετηριας **s** μεχρι να βρεθει σε αδιεξοδο(δηλαδη να μην μπορει να παει σε νέο κομβο που δεν εχουμε επισκεφθει) ή να εχει επισκεφθει ολους τους κομβους του γραφηματος. Σε κάθε νέο κομβο που περναει βαζει στην θεση του στη λιστα **saw** τηντιμη true (τον επισκεφθηκε) και ελενχει αν εχει μικροτερο βαρος από του **s** στην λιστα **min\_weight** και αν εχει ενημερωνει καταλληλα τις λιστες **min\_node** ,**min\_weight** για τον κομβο **s**.Κάθε φορα που συναντα έναν κομβο που τον εχει επισκεφθει συγκρινει την τιμη αυτου του κομβου στη λιστα **min\_weight** με την τιμη του **s** και αν είναι μικροτερη του **s** αντικαθιστα τηντιμη του **s** στις λιστες **min\_node** ,**min\_weight** με την νεα μικροτερη τιμη του κομβου.Μολις φτασει σε αδιεξοδο ή εχουμε επισκεφθει ολους τους κομβους του γραφηματος, για την αφετηρια **s** οι τιμες y=min(s) ,w(y) εχουν οριστικοποιηθει στις λιστες **min\_node** ,**min\_weight** για τον **s** και παει στην λιστα των ισχυρά συνδεδεμένων συνιστωσών βρισκει την ισχυρά συνδεδεμένη συνιστωσα οπου ανηκει ο **s** και **για ολους τους κομβους που ανηκουν στην ισχυρα συνιστωσα του** **s (αν υπαρχουν)** παει και αντικαθιστα τις τιμες τους στις λιστες **min\_node** ,**min\_weight** με τις τιμεςτου **s.**

Τελος βαζει τον **s** καιολους αυτους τους κομβους που ανηκουν στην ισχυρα συνιστωσα του **s** στην λιστα **nodes\_done** γιανα τους αφαιρεσει από την λιστα **nodes\_to\_compute** με τα τελος της εκτελεσης του **BFS.** Αν εχουμε επισκεφθει ολους τους κομβους του γραφηματος τερματριζει ο **BFS.**Διαφορετικα κάθε φορα που πεφτει σε αδιεξοδο επαναλαμβανει την παραπανω διαδικασια που περιγραφετε στο βημα 4 για νέα αφετηρια **s** τον τυχαιο κομβο που δεν εχουμε επισκεφθει από τον οποιο συνεχιζει την διερευνηση ο **BFS.**Μολις τελειωσει ο **BFS** αφαιρουμε τα στοιχεια της λιστας **nodes\_done** από την λιστα **nodes\_to\_compute (οσα υπαρχουν)** γιατι για αυτους του κομβους εχει υπολογιστει το ζητουμενο**.**Η συνολικηπολυπλοκοτητα Ο(2m+2n) m ακμες, n κομβοι.

**Ασκηση 2**

Κραταμε έναν μετρητή **count** αρχικοποιημενο στο 0 σε κάθε βημα που να μετραει ποσοι κομβοι εχουν μηδενικες εισερχομενες ακμες.Αρχικα ελενχουμε για ολους τους κομβους ποσες εισερχομενες ακμες εχουν και αν υπαρχει ακριβως ενας κομβος με μηδεν εισερχομενες ακμες τοτε συνεχιζει η εκτελεση ,διαφορετικα τερματιζει.Στην συνεχεια σβηνουμε τον κομβο που ειχε μηδεν εισερχομενες ακμες μαζι με τις εξερχομενες ακμες του και ελενχουμε μονο για τους κομβους **(**είναι οι μονοι θποψηφιοι για μηδενικες εισερχομενες μετα την διαγραφη του κομβου**)** που ειχε εξερχομενες ακμες αν (μετα την διαγραφη των ακμων) υπαρχει ακριβως ενας από αυτους(κομβος) με μηδεν εισερχομενες ακμες και αν εχει συνεχιζει με τον ιδιο τροπο.Διαφορετικα τερματιζει.**Πολυπλοκοτητα Ο(n+m).**

**Ασκηση 3**

Ο αλγοριθμος που δινεται αποτυνχανει στο να υπολογισει σωστα τις ελαφρυτατες διαδρομες από τον s σε κάθε κομβο για αρχικο μη κατευθυνομενο συνεκτικο γραφημα με ακμες με αρνητικο ακεραιο βαρος **k**. Αυτο σημαινει ότι ο αλγοριθμος κατά την μετατροπη του αρχικου μη κατευθυνομενου γραφηματος σε κατευθυνομενο γραφημα δημιουργει κυκλους αρνητικου βαρους οπου υπαρχει αρνητικη ακμη στο αρχικο γραφημα.Στην συνεχεια εκτελειται ο αλγόριθμος Bellman-Ford πανω στο νέο κατευθυνομενο γραφημα οπου για κάθε αρνητικο βαρος εχει κυκλο αρνητικου βαρους.Ο αλγόριθμος Bellman-Ford δεν υπολογιζει σωστα τις ελαφρυτατες διαδρομες όταν υπαρχουν κυκλοι αρνητικου βαρους αρα συνολικα ο αλγοριθμος δεν λειτουργει σωστα για εισοδο μη κατευθυνομενο γραφημα που εχει τουλαχιστον μια αρνητικη ακμη k.

**Ασκηση 4**

Αρχικα οριζουμε ένα λεξικο **effects** με κλειδι μια ακμη και τιμη Τrue αν η ακμη επηρεαζει την ελαφρυτατη διαδρομη από τον s->t ή False αν δεν την επηρεαζει.Αρχικοποιουμε ολες τις τιμες των ακμων στην τιμη False.

Δεδομενου ενός μη κατευθυνομενου εκτελουμε αρχικα για τον αρχικο γραφημα με ολες τις ακμες τον αλγοριθμο Dijkstra με δομη δυαδικου σωρου και αποθηκευουμε την ελαφρυτατη διαδρομη από **s -> t** .Στην συνεχεια **για κάθε ακμη**  **ι** του γραφηματος εκτελουμε Dijkstra με δομη δυαδικου σωρου στο γραφημα αγνοντας την ακμη που ειμαστε κάθε φορα(χωρις να την χρησιμοποιουμε).Συγκρινουμε την ελαφρυτατη διαδρομη που βρισκουμε από τον s στον t χωρις την **ακμη ι** που ειμαστε με αυτή που εχουμε αποθηκευσει από το αρχικο γραφημα και αν είναι διαφορετικη η ελαφρυτατη διαδρομη χωρις την ακμη που ειμαστε από την ελαφρυτατη διαδρομη που αποθηκευσαμε(με ολες τις ακμες), για την **ακμη ι** που ειμαστε παμε στην καταλληλη θεση του λεξικου **effects** και βαζουμε την τιμη Τrue.Η συνολικη πολυπλοκοτητα του αλγοριθμου είναι Ο( (m+1) *m*log*n* ) οπου m οι ακμες και n κομβους του γραφηματος.