ΑΛΓΟΡΙΘΜΟΙ - Εαρινό Εξάμηνο 2022

Ομάδα Ασκήσεων # 2

Ονομ/νυμο: ΑΛΒΙΟΝΑ ΜΑΝΤΣΟ

Αρ. Μητρώου: 3200098

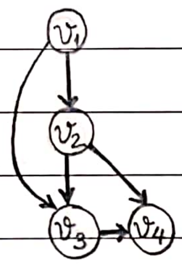
Ασκ. 1

(α) Θεωρούμε τον κατευθυνόμενο γράφο G που κατασκευάζεται ως εξής:

- Οι κορυφές του είναι το σύνολο εργασιών $V = \{v_1, v_2, \dots, v_n\}$
- Οι ακμές του είναι οι (v_j, v_i) $\forall v_j \in V_i$ όπου $i = 1, 2, \dots$

π.χ. Για $V = \{v_1, v_2, v_3, v_4\}$ και $V_1 = \emptyset$, $V_2 = \{v_1\}$, $V_3 = \{v_1, v_2\}$, $V_4 = \{v_2, v_3\}$

έχουμε G:

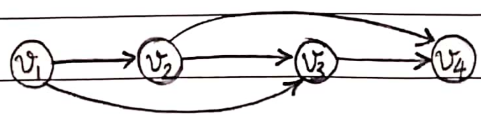


(κάθε ακμή (v_j, v_i) σημαίνει ότι η v_j πρέπει να προηγηθεί της v_i, δλδ $v_j \in V_i$)

Το σύνολο εργασιών V είναι εφικτό ανν το γράφημα G που περιγράφηκε παραπάνω είναι άκυκλο. Το τελευταίο επιτρέπει την εύρεση μιας τοπολογικής ταξινόμησης για το G, η οποία αντιστοιχεί σε μια πιθανή σειρά εκτέλεσης των εργασιών που δεν θα παραβιάζει κανέναν περιορισμό προτεραιότητας αφού:

- Μια ακμή $v_j \longrightarrow v_i$ σημαίνει ότι το v_j πρέπει να προηγηθεί του v_i
- Στην τοπολογική ταξ. όλες οι ακμές έχουν την ίδια φορά: από αριστερά προς τα δεξιά.

Για το παραπάνω παράδειγμα έχουμε:



(b) Δεδομένης της σειράς εκτέλεσης των εργασιών που υπαγορεύει η τοπολογική διάταξη του γράφου που περιγράφηκε προηγουμένως και με την υπόθεση ότι οι εργασίες μπορούν να εκτελούνται παράλληλα παρατηρούμε τα εξής:

- Η χρονική στιγμή ολοκλήρωσης μιας εργασίας $v_i \in V$ είναι η $\boxed{completed(v_i) = start(v_i) + d(v_i)}$ με $i = 1, 2, \ldots, n$ σε τοπολογική διάταξη

  χρονική στιγμή έναρξης της v_i

  χρονική διάρκεια της v_i

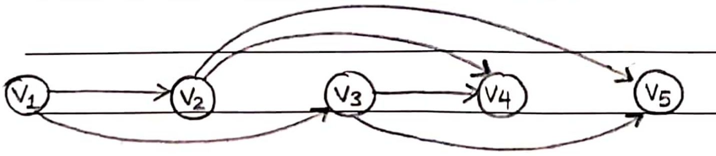
- Η χρονική στιγμή έναρξης είναι

$$\boxed{start(v_i) = \max_{v_k \in \Gamma^-(v_i)} \{completed(v_k)\}}$$

όπου θεωρούμε το max και όχι π.χ. το άθροισμα των χρόνων ολοκλήρωσης των προαπαιτούμενων εργασιών εφόσον αυτές μπορούν να εκτελεστούν παράλληλα και άρα η καθυστέρηση θα προέλθει από εκείνη που θα ολοκληρωθεί τελευταία (max).

- Ζητούμενο αποτέλεσμα: $\boxed{\max_{v \in V} \{completed(v)\}}$

Example



$d_1 = 1$ $d_2 = 3$ $d_3 = 2$ $d_4 = 4$ $d_5 = 1$

(task) : (start) + (dur) = (completed)

V_1 : $0 + 1 = 1$
V_2 : $\max\{1\} + 3 = 4$
V_3 : $\max\{1\} + 2 = 3$
V_4 : $\max\{3, 4\} + 4 = 8$
V_5 : $\max\{3, 4\} + 1 = 5$

ANSWER: 8

Algorithm

```
DFS (G)          //to obtain topological order
for v∈V in topological order:
    start(v)=0
    for u∈Γ⁻(v):
        if completed(u)>start(v)
            start(v) ← completed(u)
    completed (v) ← start (v) + d(v)

return max {completed (v)}
       v∈V
```

COMPLEXITY
O(n+m) : λόγω DFS για την τοπολογική και διπλού for-loop

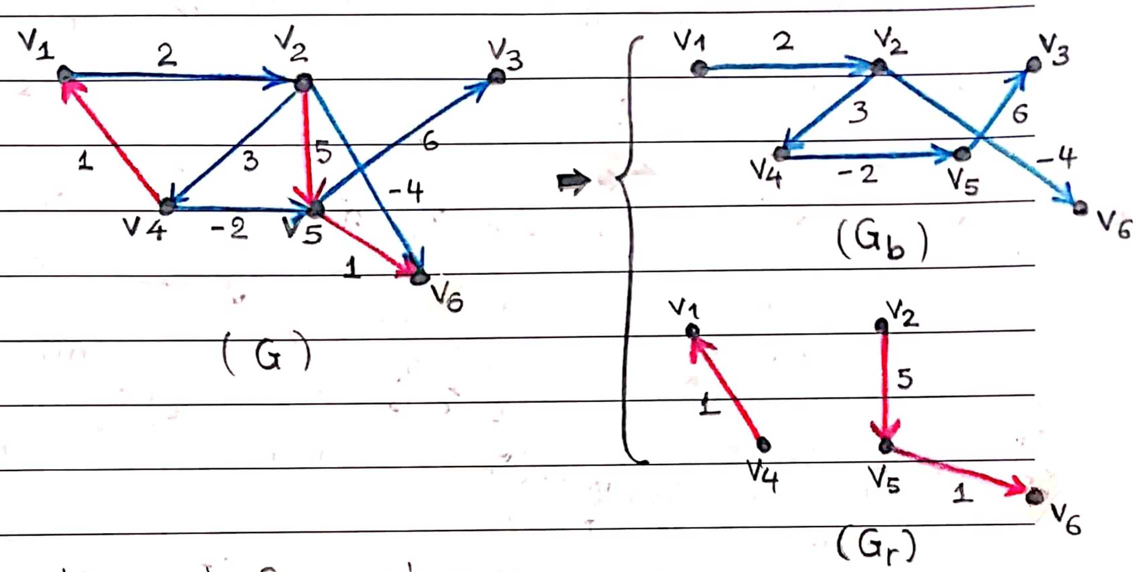
Ασκ 2.

Έστω G ο γράφος που περιγράφεται στην άσκηση
Θεωρούμε τα υπογραφήματα G_r, G_b όπου:

→ $G_r = (V_r, E_r)$: $E_r = \{e \in V \,/\, e$ είναι <u>κόκκινη</u> ακμή$\} \subseteq E$
$V_r = \{v \in V \,/\, \exists e \in E_r$ που να ακουμπάει στο $v\} \subseteq V$

→ $G_b = (V_b, E_b)$: $E_b = \{e \in V \,/\, e$ είναι <u>μπλε</u> ακμή$\} \subseteq E$
$V_b = \{v \in V \,/\, \exists e \in E_b$ που να ακουμπάει στο $v\} \subseteq V$

π.χ.



Δεδομένης της υπόθεσης ότι δεν υπάρχουν μονοχρωματικοί κύκλοι, τα γραφήματα G_r, G_b δεν έχουν κύκλους.
Παρατηρούμε επίσης ότι τα G_r, G_b που προκύπτουν:

1) Θα έχουν κάποιες κορυφές κοινές, ενώ άλλες μπορεί να εμφανίζονται μόνο στο ένα εκ των 2.
2) Μπορεί να μην είναι συνεκτικά γραφήματα (όπως συμβαίνει παραπάνω με το G_r)

Δίνονται οι κόμβοι s, t και αναζητούμε το συντομότερο μονοπάτι από s σε t που αλλάζει χρώμα ακριβώς μια φορά <u>από μπλε σε κόκκινο</u>. Άρα ο s πρέπει να βρίσκεται τουλάχιστον στο γράφημα G_b και ο t τουλάχιστον στο γράφημα G_r. Επίσης, αν οι G_b, G_r έχουν περισσότερες από μια συνιστώσες, οι κόμβοι s, t θα ανήκουν αντίστοιχα σε μια από αυτές (διαφορετικά θα μπορούσαμε να ενώσουμε τις συνιστώσες σε μια)

Λαμβάνοντας υπόψιν όλα τα παραπάνω:

Βήμα 1 → Εντοπίζουμε τη συνιστώσα του G_b που περιλαμβάνει τον s (αν δεν υπάρχει, τότε δεν υπάρχει μονοπάτι s⇝t που πληροί τις προδιαγραφές)

Βήμα 2 → Εντοπίζουμε τη συνιστώσα του G_r που περιλαμβάνει τον t (αν δεν υπάρχει, και πάλι δεν υπάρχει μονοπάτι s⇝t)

Βήμα 3 → Λόγω της παρατήρησης (1) οι συνιστώσες αυτές θα είναι ακυκλικές. Οπότε βρίσκουμε την τοπολογική διάταξη για την καθε μία.

Βήμα 4 → Στη συνέχεια, δεδομένων των 2 τοπολογικών διατάξεων:

G_b { 1) Βρίσκουμε το συντομότερο μονοπάτι από τον s προς κάθε κόμβο που έπεται του s στην τοπολογική διάταξη και συγκρατούμε τους κόμβους αυτούς μαζί με την συντομότερη απόστασή τους από τον s σε ένα σύνολο S_b

G_r { 2) Αντιστρέφουμε τις ακμές στην άλλη τοπολογική διάταξη (κόκκινη, περιέχει t) και κάνουμε το ίδιο για τον t προς τους κόμβους που έπονται του t στην αντιστροφή και άρα προηγούνται στο αρχικό γράφημα. → σύνολο S_r

Βήμα 5 → Για τους κοινούς κόμβους των S_b, S_r ($v \in S_b$ και $v \in S_r$) παίρνουμε το μονοπάτι που ελαχιστοποιεί το άθροισμα $(s \rightsquigarrow v) + (v \leftsquigarrow t) = (s \rightsquigarrow v) + (v \rightsquigarrow t)$ και το οποίο έχει μπλε ακμές στο τμήμα $s \sim v$ και κόκκινες στο $v \sim t$, δλδ αλλάζει χρώμα ακριβώς 1 φορά, όπως ζητείται.

-Ανάλυση

→ Για τα βήματα 1,2 αρκεί να παρατηρήσουμε ότι στη συνέχεια θα εργαστούμε μόνο με τους κόμβους που είναι προσβάσιμοι από τον s με μπλε ακμές, και με τους κόμβους που καταλήγουν στον t με κόκκινες ακμές. Τροποποιώντας ελάχιστα τον αλγόριθμο DFS ώστε με αφετηρία τον s να ανακαλύπτει

τους κόμβους με την εξής συνθήκη:
'for each $v \in (\Gamma^+(u))$ and $e=(u,v)$ is blue'

μπορούμε να πάρουμε το τμήμα της συνιστώσας του G_b που μας ενδιαφέρει, δλδ τους κόμβους που θα έπονται του s στην τοπολογική διάταξη (βήμα 4) με τις αντίστοιχες ακμές

Όμοια, παίρνοντας το $G^{reversed}$ και εφαρμόζοντας DFS από τον t με συνθήκη: 'for each $v \in (\Gamma^+(u))$ and $e=(u,v)$ is red'
παίρνουμε το τμήμα της συνιστώσας του G_r που μας ενδιαφέρει, δλδ τους κόμβους που έπονται του t στο reversed και άρα προηγούνται του t στην τοπολογική διάταξη (βήμα 4) με τις αντίστοιχες ακμές

Για τα αποτελέσματα που έδωσε η εφαρμογή της παραλλαγμένης DFS βρίσκουμε τις 2 τοπολογικές διατάξεις κατά τα γνωστά (μάλιστα έχουμε ήδη τα post)

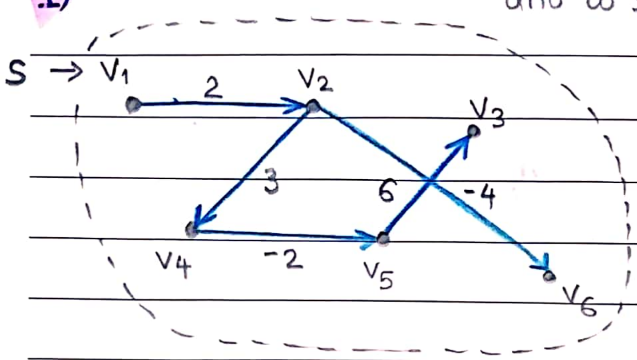
Εφαρμόζουμε δυναμικό προγραμματισμό για να βρούμε τις ελάχιστες αποστάσεις 2 φορές, μία φορά για τον s επί της πρώτης (μπλε) τοπολογικής και μια για τον t επί της δεύτερης (κόκκινης)
↳ Για τους κοινούς κόμβους v των τοπολογικών παίρνουμε το min{(s⇝v)+(v⇝t)}

Συνοπτικά:
1) DFS - με red από τον s → δέντρο του s (κόκκινο)
2) DFS - με blue στον $G^{reversed}$ από τον t → δέντρο του t (μπλε)
3) Τοπολογική διάταξη στο δέντρο του s και στο δέντρο του t (έχουμε τα post από 2,3)
4) Εύρεση συντομότερου μονοπατιού από τον s στην τοπολογική και αντίστοιχα από τον t στη δική του τοπολογική.
Κρατάμε τα σύνολα S_b, S_r που περιγράφτηκαν προηγουμένως
5) Για τα κοινά στοιχεία $v \in S_b$ και $v \in S_r$ παίρνουμε αυτό για το οποίο $dist(s,v) + dist(v,t)$ είναι ελάχιστο
6) Αν δεν βρέθηκε v στο 5, δεν υπάρχει το ζητούμενο μονοπάτι.
Αλλιώς είναι το s⇝v⇝t με κόστος $dist(s,v)+dist(v,t)$

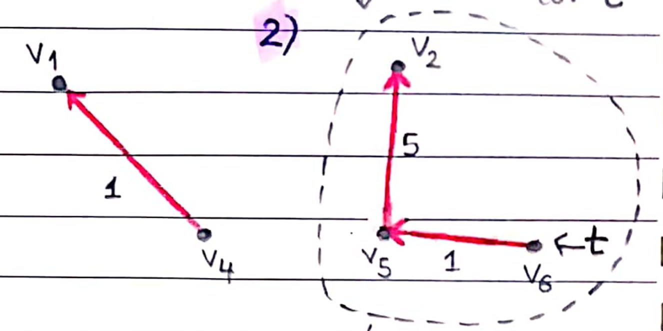
COMPLEXITY
$O(m+n)$

Εφαρμογή στο αρχικό παράδειγμα:

1) DFS στον αρχικό G με μπλε από το s



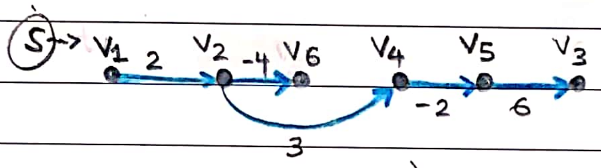
2) DFS στον $G^{reversed}$ με κόκκινο από τον t

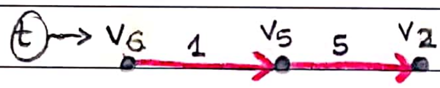


Topological order ανάλογα με το post-visit

Topological order ανάλογα με το post-visit

3)





Shortest path Alg (source: s)

Shortest path Alg (source: t)

4)

s → v_1	v_2	v_6	v_4	v_5	v_3
0	∞	∞	∞	∞	∞
	2	∞	∞	∞	∞
		-2	∞	∞	∞
			5	∞	∞
				3	∞
					9

t → v_6	v_5	v_2
0	∞	∞
	1	∞
		6

S_b

S_r

$S_b = \{(v_2, 2), (v_6, -2), (v_4, 5), (v_5, 3), (v_3, 9)\}$

$S_r = \{(v_5, 1), (v_2, 6)\}$

5)

$(S_b \cap S_r)_v = \{v_5, v_2\}$

6) Άρα: 2 πιθανά μονοπάτια: $s \to v_5 \to v_6$: cost $3+1=4$ ← min!

$s \to v_2 \to v_6$: cost $2+6=8$

$G=(V,E)$, $|V|=n$, $|E|=m$

Ασκ3. Οι πιθανές αλλαγές που μπορούν να συμβούν είναι δύο ειδών και θα εξετάσουμε την καθεμία ξεχωριστά:

"Input: G,
T,
<η αλλαγή>"

1) Διαγραφή ακμής e

Διακρίνουμε τις εξής περιπτώσεις:

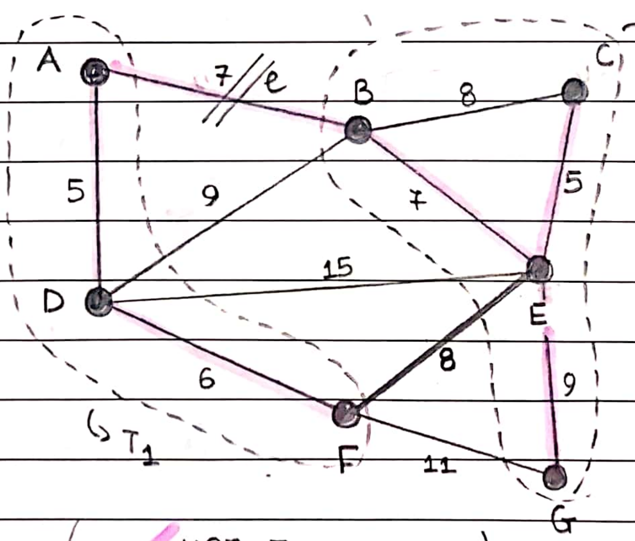
i) $e \notin T$.

Σε αυτήν την περίπτωση το T δεν χρειάζεται να μεταβληθεί, αφού αν υποθετικά εφαρμοζόταν ένας αλγόριθμος για την εξολοκλήρου κατασκευή του "νέου" T (T') εφόσον ούτε στο T είχε επιλεγεί η e, θα καταλήγαμε στο $T'=T$.

COMPLEXITY
O

ii) $e \in T$

Στην περίπτωση αυτήν τίθεται το ζήτημα του αν η διαγραφή της e αφήνει το γράφημα συνεκτικό, αλλά υποθέτουμε πως το αφήνει, ώστε να εξακολουθεί να έχει νόημα η εύρεση του T'. Με την αφαίρεση της e, το T αποτελείται από δύο υποδέντρα T_1, T_2 που πρέπει να ξανασυνδεθούν.



Έχουμε την τομή $C=\{(u,v) / u \in V_{T_1}, v \in V - V_{T_1}\}$

$\Rightarrow C=\{(u,v) / u \in V_{T_1}, v \in V_{T_2}\}$

→ Για το διπλανό παράδειγμα είναι $C=\{DB, DE, FE, FG\}$.

MST T
// e: φεύγει (η AB)
/ e': στο T' (η FE)

Από το cut property, Αρκεί να επιλέξουμε την ακμή του C με το μικρότερο βάρος (e') για να συμπληρώσουμε το T'.

→ εδώ την FE

Οπότε τελικά το MST γίνεται από $T \rightarrow T'$ ως εξής:

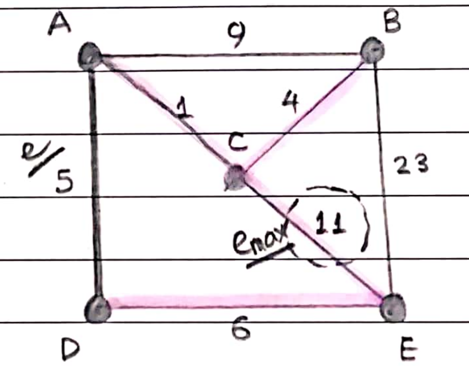
$$T' = (V, E_T - \{e\} \cup \{e'\})$$

COMPLEXITY
$O(m)$
(διατρέχουμε τις ακμές για να προσδιορίσουμε το C και βρίσκουμε τη min)

2) <u>Προσθήκη ακμής e με βάρος w</u>

Έστω πως προσθέτουμε την e στο T. Αφού το T είναι δέντρο, με την προσθήκη της e δημιουργείται κύκλος.
Από το cycle property, είναι βέβαιο πως η ακμή μέγιστου βάρους του κύκλου που δημιουργήθηκε δεν θα συμμετέχει στο "νέο" MST T'. Αφαιρούμε λοιπόν την ακμή αυτή (e_{max}) και παίρνουμε το T'.
(Το T' είναι συνδετικό δέντρο γιατί
- έχει κόμβους το V
- έχει $\underbrace{|V-1|}_{(T)} + \underbrace{1}_{(e)} - \underbrace{1}_{(e_{max})} = |V-1|$ ακμές και κανέναν κύκλο.)

• Παρατηρούμε το εξής
Αν $e = e_{max}$ τότε $T' = T$
αλλιώς $T' \neq T$



T

COMPLEXITY:

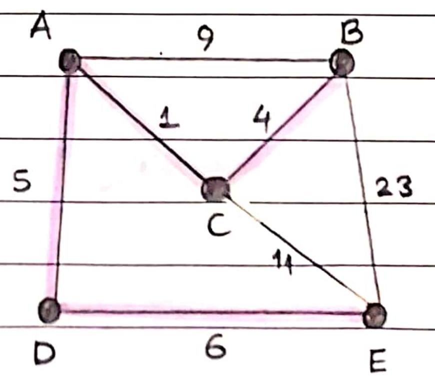
Ο κύκλος εντοπίζεται με DFS στο δέντρο της οποίας θα υπάρχει back ακμή.

$O(m+n) \Leftarrow$ Οι υποψήφιες προς έλεγχο ακμές είναι η back ακμή (u,v) και όλες οι $v \rightarrow \cdots \rightarrow u$ ακμές του δέντρου

$O(m) \Leftarrow$ Βρίσκουμε την ακμή με το μέγιστο βάρος

$\Downarrow$

$O(n+m) = O(m)$ (αφού είναι συνεκτικός ο γράφος)



T'

(εδώ $e \neq e_{max}$, οπότε $T \neq T'$)

Δεδομένος γράφος $G=(V,E)$. $|V|=n$, $|E|=m$

Ασκ. 4. Έστω 2k-κλίκες το περιγραφόμενο πρόβλημα.

Για να δείξουμε ότι το 2k-κλίκες είναι NP-πλήρες αρκεί να δείξουμε:

1) 2k-κλίκες $\in NP$

2) Όλα τα NP προβλήματα ανάγονται πολυωνυμικά στο 2k-κλίκες.

1) 2k-κλίκες $\in NP$ γιατί:

Με πιστοποιητικό (certificate): 2 σύνολα κόμβων V_1, V_2 ...

... ο επαληθευτής (verifier) εκτελεί τους ελέγχους:

a. $V_1 \subseteq V$ και $V_2 \subseteq V$ $\rightarrow O(n)$

b. $|V_1| \geq k$ και $|V_2| \geq k$ $\rightarrow O(n)$

c. $V_1 \cap V_2 = \emptyset$ $\rightarrow O(n^2)$

d. ελέγχει για όλα τα ζεύγη (u,v) στο V_1 αν υπάρχει ακμή $e=(u,v) \in E$ $\rightarrow O(n^2)$

O(poly)

Άρα ένα δεδομένο πιστοποιητικό (V_1, V_2) πολυωνυμικό, μπορεί να επαληθευτεί σε πολυωνυμικό χρόνο. $\Longrightarrow$ 2k-κλίκες $\in NP$

2) Γνωρίζουμε ότι το πρόβλημα της ύπαρξης k-κλίκας (έστω πρόβλημα k-κλίκα) είναι NP-complete. Αυτό σημαίνει ότι για κάθε πρόβλημα $\Pi \in NP$ ισχύει $\Pi \leq_p$ k-κλίκα.

Συνεπώς αν δείξουμε ότι είναι εφικτή η αναγωγή k-κλίκα $\leq_p$ 2k-κλίκες θα έχουμε δείξει λόγω:

i) της μεταβατικότητας των αναγωγών ($\Pi \leq_p$ k-κλίκα $\leq_p$ 2k-κλίκες)

ii) του γεγονότος ότι η σύνθεση πολυωνύμων είναι πολυώνυμο

... ότι 2k-κλίκες είναι NP-complete

k-κλίκα $\leq_p$ 2k-κλίκες

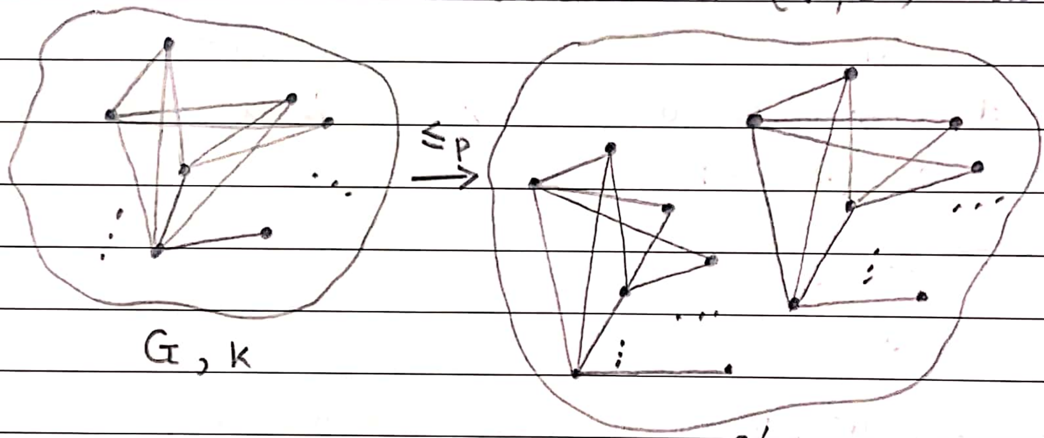
Δεδομένου του γράφου $G=(V,E)$ και $k \in \mathbb{N}_{\geq 2}$ ως είσοδο για το πρόβλημα k-κλίκα κάνουμε την εξής αναγωγή:

Κατασκευάζουμε τον γράφο $G' = (\underbrace{V \cup V_{copy}}_{V'}, \underbrace{E \cup E_{copy}}_{E'})$

↳ που θα έχει ως κόμβους τους κόμβους του G (V) καθώς και το σύνολο των κόμβων που προκύπτουν ως "αντίγραφα" των πρώτων (V_{copy}), δλδ πρακτικά για κάθε κόμβο στον G δημιουργούμε έναν ακόμη στον G'

↳ θα έχει ως ακμές τις ακμές του G (E) και τις ομόλογες ακμές επί των κόμβων του V_{copy} (E_{copy})

Η είσοδος του 2k-κλίκες θα είναι $G'=(V',E')$ και k (ίδιο με το αρχικό k).



G , k

G', k

Προφανώς ο μετασχηματισμός αυτός μπορεί να γίνει σε πολυωνυμικό χρόνο.

Αρκεί τώρα να δείξουμε ότι $\boxed{\text{instance-}\langle G,k\rangle \in \text{k-κλίκα} \iff \text{instance}\langle G',k\rangle \in \text{2k-κλίκες}}$

I. instance-$\langle G,k\rangle \in$ k-κλίκα $\Longrightarrow$ Instance-$\langle G',k\rangle \in$ 2k-κλίκες.

▶ Αν ο G έχει k-κλίκα, σημαίνει ότι $\exists V_c \subseteq V$ που είναι οι κόμβοι της k-κλίκας. Τότε ο G' θα έχει σίγουρα 2 k-κλίκες:

1) Η πρώτη θα είναι η $V_c \subseteq V \subseteq V'$

2) Η δεύτερη θα είναι οι ομόλογοι κόμβοι των V_c στο σύνολο V_{copy}, έστω σύνολο V_c'

Δλδ κλίκα η $V_c' \subseteq V_{copy} \subseteq V'$

ΙΙ. instance $-\langle G', k\rangle \in$ 2k-κλίκες $\Longrightarrow$ instance $-\langle G, k\rangle \in$ k-κλίκα

▶ Έστω V_{c_1}, V_{c_2} οι 2 k-κλίκες με $V_{c_1} \cap V_{c_2} = \emptyset$.

Υπάρχουν οι εξής περιπτώσεις

1) $V_{c_1}, V_{c_2} \subseteq V \subseteq V'$ και άρα G' έχει 2 k-κλίκες και επιπλέον άλλες 2 ως υποσύνολα του $V_{copy} \subseteq V'$ (που θα είναι δλδ. οι ομόλογες των V_{c_1}, V_{c_2} επί του συνόλου V_{copy}).

Άρα ο G' θα έχει 4 k-κλίκες και αφού $V_{c_1}, V_{c_2} \subseteq V$, <u>ο G έχει k-κλίκα</u>, και μάλιστα 2

2) $\left.\begin{array}{l} V_{c_1} \subseteq V \\ V_{c_2} \subseteq V_{copy} \end{array}\right\}$

Αφού $V_{c_1} \subseteq V$ ο G έχει 1 k-κλίκα άρα <u>ο G έχει k-κλίκα</u>

Και το ζητούμενο αποδείχθηκε