



ΗΥ - 280 Θεωρία Υπολογισμού

Διάλεξη 18

Κωνσταντίνος Βάρσος

Τμήμα Επιστήμης Υπολογιστών – Πανεπιστήμιο Κρήτης

Χειμερινό εξάμηνο 2025 - 26

Πολυπλοκότητα

Τι είδαμε στην Διάλεξη 17

- Εισαγωγή στην Πολυπλοκότητα.
- Μερικά χρήσιμα εργαλεία από την Ασυμπτωτική Ανάλυση.
- Σχέσεις Πολυπλοκότητας μεταξύ Μοντέλων.

Ανάλυση Αλγορίθμων

Ορισμός. Έστω TM M , **χρόνος υπολογισμού** ή **χρονική πολυπλοκότητα** της M είναι μια συνάρτηση $f : \mathbb{N} \rightarrow \mathbb{N}$ τ.ω. για κάθε είσοδο x με $|x| = n$, η M τερματίζει μέσα σε $f(n)$ αριθμό βημάτων. [συνοπτικά θα την αναφέρουμε ως TM χρόνου $f(n)$]

■ Γενικά, επιλέγουμε την καλύτερη δυνατή συνάρτηση $f(n)$ όταν περιγράφουμε την χρονική πολυπλοκότητα μιας TM.

Ορισμός. Έστω $t : \mathbb{N} \rightarrow \mathbb{R}_{\geq 0}$ οποιαδήποτε συνάρτηση. Ορίζουμε την **κλάση χρονικής πολυπλοκότητας** $\text{TIME}(t(n))$ τη συλλογή όλων των γλωσσών που μπορούν να διαγνωστούν από μια TM χρόνου $O(t(n))$.

■ Για $t : \mathbb{N} \rightarrow \mathbb{R}_{\geq 0}$:

$$\text{TIME}(t(n)) = \{L \mid L = L(M) \text{ για TM } M \text{ με } O(t(n)) \text{ χρονική πολυπλοκότητα}\}$$

Κλάση P

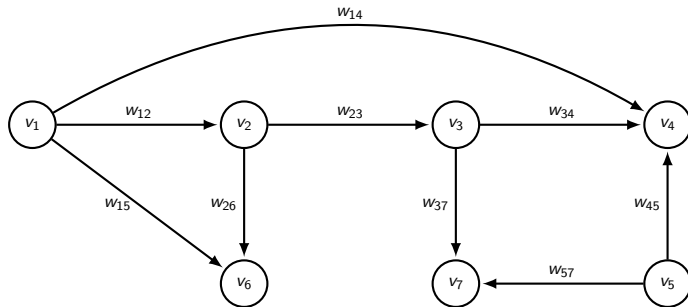
- Ορίζουμε ως **P** την **κλάση όλων των διαγνώσιμων σε πολυωνυμικό χρόνο γλωσσών**,

$$P = \bigcup_k TIME(n^k)$$

- Η κλάση P είναι αναλλοίωτη για όλα τα αιτιοκρατικά ντετερμινιστικά μοντέλα.
- Αντιστοιχεί, όχι επακριβώς, στα πρακτικώς υπολογίσιμα προβλήματα.
- Η χρονική πολυπλοκότητα μιας TM είναι μια ιδιότητα της μηχανής και όχι του προβλήματος [διαφορετικές μηχανές συμπεριφέρονται διαφορετικά για το ίδιο πρόβλημα]

π.χ., Πολλαπλασιαμός πινάκων, διάσχιση γράφου, όλες οι γλώσσες χωρίς συμφραζόμενα, υπολογισμός ορίζουσας πίνακα μέσω της μεθόδου Gauss, κτλ.

Κλάση P



■ Ορίζουμε την κλάση προβλημάτων *PATH*, ως εξής

$$PATH := \{ \langle G, s, t \rangle \mid G \text{ είναι κατευθυνόμενο γράφημα με μονοπάτι από το } s \text{ στο } t \}$$

Κλάση P

Θεώρημα. $PATH \in P$.

Απόδειξη. Κατασκευάζουμε TM M η οποία να διαγιγνώσκει το $PATH$ εκτελώντας τις παρακάτω ενέργειες σε είσοδο $\langle G, s, t \rangle$.

$M =$ “Για είσοδο $\langle G, s, t \rangle$:

1. Σημείωσε την κορυφή s .
2. Για κάθε σημειωμένη κορυφή x , διέτρεξε τον G και σημείωσε όλες τις μη-σημειωμένες κορυφές που γειτνιάζουν με την x .
3. Επανάλαβε το προηγούμενο βήμα έως ότου δεν σημειωθεί κάποια καινούργια ακμή.
4. Αποδέχεται αν η t σημειωθεί, αλλιώς απορρίπτει.”

Κλάση P

Θεώρημα. $PATH \in P$.

Απόδειξη. Κατασκευάζουμε TM M η οποία να διαγιγνώσκει το $PATH$ εκτελώντας τις παρακάτω ενέργειες σε είσοδο $\langle G, s, t \rangle$.

$M =$ “Για είσοδο $\langle G, s, t \rangle$:

1. Σημειώνει την κορυφή s .
2. Για κάθε σημειωμένη κορυφή x , διέτρεξε τον G και σημείωσε όλες τις μη-σημειωμένες κορυφές που γειτνιάζουν με την x . [$(\leq n$ επαναλήψεις) $\times \leq O(n^2)$ βήματα)]
3. Επανάλαβε το προηγούμενο βήμα έως ότου δεν σημειωθεί κάποια καινούργια ακμή.
[$\leq n$ επαναλήψεις]
4. Αποδέχεται αν η t σημειωθεί, αλλιώς απορρίπτει.”

■ Συνολικά, $O(n^4)$ βήματα. [για να δείξουμε πολυωνυμικό χρόνο θα πρέπει τόσο κάθε βήμα όσο και ο αριθμός των βημάτων να έχουν πολυωνυμικό μέγεθος.]

Κλάση P

- Ορίζουμε την κλάση προβλημάτων *RELPRIME* (Relative Primes), ως εξής

$$RELPRIME := \{ \langle x, y \rangle \mid x, y \text{ είναι αμοιβαίως πρώτοι αριθμοί} \}$$

π.χ., Οι αριθμοί 8 και 9 είναι αμοιβαίως πρώτοι αριθμοί. [$\mu\kappa\delta(8, 9) = 1$]

- Οι αριθμοί x, y θα κωδικοποιηθούν σε δυαδική αναπαράσταση.

- Το μήκος της αναπαράστασης $\langle x, y \rangle$ είναι $O(\log(x + y))$.

- Θα βοηθούσε κάποιος εξαντλητικός αλγόριθμος; [γιατί;]

Κλάση P

■ Ευκλείδειος αλγόριθμος.

Input x, y , όπου x, y φυσικοί αριθμοί :

$z \leftarrow x$

while $z \neq 0$:

$z \leftarrow x \bmod(y)$

$x \leftarrow y$

$y \leftarrow z$

Output x .

π.χ., $x = 48$, $y = 18$: $12 = 48 \bmod(18)$, $6 = 18 \bmod(12)$, $0 = 12 \bmod(6)$. Άρα ο αλγόριθμος επιστρέφει 6.

Κλάση P

■ Ευκλείδιος αλγόριθμος.

$E =$ "Για είσοδο $\langle x, y \rangle$, όπου x, y φυσικοί αριθμοί στο δυαδικό σύστημα:

1. Επαναλαμβάνει έως ότου $y = 0$:
2. Αναθέτει στο x την τιμή $x \bmod y$.
3. Εναλλάσσει τις τιμές των x και y .
4. Επιστρέφει x ."

■ Το 2. μειώνει το x κατά το ήμισυ. Επίσης, λόγω 3. και το y μειώνεται κατά το ήμισυ.

■ Το μικρότερο πλήθος εκτελέσεων των 2. και 3. είναι το $\min\{2 \log_2 x, 2 \log_2 y\} < \log_2(x + y)$.

■ Ο E για είσοδο $\langle x, y \rangle$ τρέχει σε πολυωνικό χρόνο, $O(\log(x + y))$. Επομένως, η χρονική πολυπλοκότητα του E είναι $O(n)$.

Κλάση P

■ $RELPRIME \in P$.

Θα κατασκευάσουμε έναν αλγόριθμο R ο οποίος θα καλεί τον E ως υπορουτίνα, ώστε να λύνει το πρόβλημα των αμοιβαία πρώτων αριθμών.

$R =$ “Για είσοδο $\langle x, y \rangle$, όπου x, y φυσικοί αριθμοί στο δυαδικό σύστημα:

1. Εκτέλεσε E για είσοδο $\langle x, y \rangle$.
2. Εάν η έξοδος του E είναι 1, αποδέχεται.

Διαφορετικά, απορρίπτει.”

■ Ο R καλεί τον E , άρα εκτελείται σε πολυωνυμικό χρόνο.

Κλάση P

Θεώρημα. Κάθε $CFL \in P$.

Έστω L CFL. Άρα υπάρχει CFG G σε CNF τ.ω. $L(G) = L$. [Κάθε παραγωγή απαιτεί $2n - 1$ βήματα, όπου n το μήκος της w]

Για κάθε είσοδο $w = w_1 w_2 \dots w_n$ θέλουμε να διαγνώσουμε, είτε θετικά είτε αρνητικά, σε πολυωνυμικό χρόνο αν $w \in L(G)$.

Μια απλοϊκή προσέγγιση, απαρίθμηση όλων των παραγωγών μήκους $2n - 1$. [Brute-force algorithm]

$M =$ "Σε είσοδο $w = w_1 w_2 \dots w_n$:

1. Κατασκεύασε όλες τις δυνατές παραγωγές της αντίστοιχης CNF.
2. Αν κάποια παραγωγή παράγει την w , αποδέχεται.
Αλλιώς, απορρίπτει.'

Κλάση P

Θεώρημα. Κάθε $\text{CFL} \in \text{P}$.

Έστω L CFL. Άρα υπάρχει CFG G σε CNF τ.ω. $L(G) = L$. [Κάθε παραγωγή απαιτεί $2n - 1$ βήματα, όπου n το μήκος της w]

Για κάθε είσοδο $w = w_1 w_2 \dots w_n$ θέλουμε να διαγνώσουμε, είτε θετικά είτε αρνητικά, σε πολυωνυμικό χρόνο αν $w \in L(G)$.

Μια απλοϊκή προσέγγιση, απαρίθμηση όλων των παραγωγών μήκους $2n - 1$. [Brute-force algorithm]

Όχι καλή πρακτική, θα χρειαστεί εκθετικό χρόνο.

Κλάση P

Θεώρημα. Κάθε $CFL \in P$.

Απόδειξη. [Ιδέα] Θα χρησιμοποιήσουμε δυναμικό προγραμματισμό. [τεχνική σύμφωνα με την οποία χρησιμοποιούμε υποπροβλήματα ώστε να λύσουμε το κυρίως πρόβλημα.]

Αρχικά ορίζουμε το εξής εργαλείο: Για είσοδο $w = w_1 w_2 \dots w_n$ δημιουργούμε τις περιστασιακές μεταβλητές $table(i, j)$ για $1 \leq i \leq j \leq n$ τ.ω.

$$A \in table(i, j) \iff \left(A \xRightarrow{*} w_i w_{i+1} \dots w_j \right).$$

...

Κλάση P

π.χ., Δίνεται η CNF G

$$S \rightarrow AA \mid \varepsilon, \quad A \rightarrow BB \mid AB \mid a, \quad B \rightarrow BA \mid b.$$

Υπολογίστε τις $table(i, j)$ για κάθε $1 \leq i \leq j \leq 4$ έτσι ώστε να δείξετε ότι $abba \in L(G)$.

Λύση. Κατασκευάζουμε τον εξής αλγόριθμο.

Αν $A \rightarrow w_i \in G$, προσθέτει $A \in table(i, i)$.

Επομένως για την $table$ έχουμε τα εξής:

	1	2	3	4
1	A			
2		B		
3			B	
4				A

...

Κλάση P

π.χ., Δίνεται η CNF G

$$S \rightarrow AA \mid \varepsilon, \quad A \rightarrow BB \mid AB \mid a, \quad B \rightarrow BA \mid b.$$

Υπολογίστε τις $table(i, j)$ για κάθε $1 \leq i \leq j \leq 4$ έτσι ώστε να δείξετε ότι $abba \in L(G)$.

Λύση.(συνέχεια) Χρησιμοποιούμε την επανάληψη

Για $\ell = 2$ έχουμε,

Για $\ell = 1, \dots, n$:

$\forall i, k$ τ.ω. $1 \leq i \leq k \leq i + \ell - 1 \leq n$: $[j = i + \ell - 1]$

$\forall A \rightarrow BC \in G$:

Αν $B \in table(i, k)$ και $C \in table(k + 1, j)$

τότε πρόσθεσε $A \in table(i, j)$.

	1	2	3	4
1	A	A		
2		B	A	
3			B	B
4				A

...

Κλάση P

π.χ., Δίνεται η CNF G

$$S \rightarrow AA \mid \varepsilon, \quad A \rightarrow BB \mid AB \mid a, \quad B \rightarrow BA \mid b.$$

Υπολογίστε τις $table(i, j)$ για κάθε $1 \leq i \leq j \leq 4$ έτσι ώστε να δείξετε ότι $abba \in L(G)$.

Λύση.(συνέχεια)

Για $\ell = 3$ έχουμε,

	1	2	3	4
1	A	A	S, A	
2		B	A	A, S
3			B	B
4				A

Για $\ell = 4$ έχουμε,

	1	2	3	4
1	A	A	S, A	S, A
2		B	A	A, S
3			B	B
4				A

Η επανάληψη ολοκληρώθηκε και ελέγχουμε: αν $S \in table(1, n)$. Αν ισχύει, τότε $abba \in L(G)$.

□

Κλάση P

Απόδειξη. (συνέχεια) Κατασκευάζουμε τον εξής αλγόριθμο.

$T =$ “Για είσοδο $w_1 w_2 \dots w_n$:

1. Αν $w = \varepsilon$, τότε αν $S \rightarrow \varepsilon \in G$, αποδέχεται. Αλλιώς, απορρίπτει.
2. Για $i = 1, \dots, n$:
3. Αν $A \rightarrow w_i \in G$, προσθέτει $A \in table(i, i)$.
4. Για $\ell = 1, \dots, n$:
5. $\forall i, k$ τ.ω. $1 \leq i \leq k \leq i + \ell - 1 \leq n$: $[j = i + \ell - 1]$
6. $\forall A \rightarrow BC \in G$:
7. Αν $B \in table(i, k)$ και $C \in table(k + 1, j)$ τότε πρόσθεσε $A \in table(i, j)$.
8. Αν $S \in table(1, n)$, αποδέχεται. Αλλιώς, απορρίπτει.” $\square$

Κλάση P

Απόδειξη. (συνέχεια) Κατασκευάζουμε τον εξής αλγόριθμο.

$T =$ “Για είσοδο $w_1 w_2 \dots w_n$:

1. Αν $w = \varepsilon$, τότε αν $S \rightarrow \varepsilon \in G$, **αποδέχεται**. Αλλιώς, **απορρίπτει**.
2. Για $i = 1, \dots, n$: $[O(n)]$
3. Αν $A \rightarrow w_i \in G$, προσθέτει $A \in table(i, i)$.
4. Για $\ell = 1, \dots, n$: $[O(n)]$
5. $\forall i, k$ τ.ω. $1 \leq i \leq k \leq i + \ell - 1 \leq n$: $[O(n^2)]$
6. $\forall A \rightarrow BC \in G$:
7. Αν $B \in table(i, k)$ και $C \in table(k + 1, j)$ τότε πρόσθεσε $A \in table(i, j)$.
8. Αν $S \in table(1, n)$, **αποδέχεται**. Αλλιώς, **απορρίπτει**. □

Κλάση P

Απόδειξη. (συνέχεια) Κατασκευάζουμε τον εξής αλγόριθμο.

$T =$ “Για είσοδο $w_1 w_2 \dots w_n$:

1. Αν $w = \varepsilon$, τότε αν $S \rightarrow \varepsilon \in G$, **αποδέχεται**. Αλλιώς, **απορρίπτει**.
2. Για $i = 1, \dots, n$: $[O(n)]$
3. Αν $A \rightarrow w_i \in G$, προσθέτει $A \in table(i, i)$.
4. Για $\ell = 1, \dots, n$: $[O(n)]$
5. $\forall i, k$ τ.ω. $1 \leq i \leq k \leq i + \ell - 1 \leq n$: $[O(n^2)]$
6. $\forall A \rightarrow BC \in G$:
7. Αν $B \in table(i, k)$ και $C \in table(k + 1, j)$ τότε πρόσθεσε $A \in table(i, j)$.
8. Αν $S \in table(1, n)$, **αποδέχεται**. Αλλιώς, **απορρίπτει**. □

■ Άρα η διάγνωση των CFL απαιτεί πολυωνυμικό χρόνο, $O(n^3)$.

Κλάση P

Θεώρημα. Η κλάση **P** είναι κλειστή ως προς τις πράξεις της τομής, της ένωσης, του συμπληρώματος, της παράθεσης και της Kleene star.

■ Με άλλα λόγια, αν A, B δύο διαγνώσιμες σε πολυωνυμικό χρόνο γλώσσες, τότε και οι γλώσσες

$$A \cap B, \quad A \cup B, \quad \overline{A}, \quad AB, \quad BA, \quad A^*$$

είναι διαγνώσιμες σε πολυωνυμικό χρόνο επίσης.

Κλάση P

Θεώρημα. Αν A, B δύο διαγνώσιμες σε πολυωνυμικό χρόνο γλώσσες, τότε και η γλώσσα $A \cup B$ είναι διαγνώσιμη σε πολυωνυμικό χρόνο. [$\text{Αν } A, B \in \mathbf{P} \Rightarrow A \cup B \in \mathbf{P}$]

Απόδειξη. Εφόσον $A, B \in \mathbf{P}$, τότε

1. $\exists$ διαγνώστης M_A για την A ο οποίος να διαγιγνώσκει σε υπολογιστικό χρόνο $O(n^k)$ για κάποιο $k \in \mathbb{N}$.
2. $\exists$ διαγνώστης M_B για την B ο οποίος να διαγιγνώσκει σε υπολογιστικό χρόνο $O(n^\ell)$ για κάποιο $\ell \in \mathbb{N}$.

...

Κλάση P

Απόδειξη. (συνέχεια) Κατασκευάζουμε μια TM 2-ταινιών για την διάγνωση της $A \cup B$ ως εξής:

M = “Για είσοδο w :

1. Αντιγράφει w , στην 2η ταινία.
2. Στην 1η ταινία εκτελεί M_A για x .
3. Αν η M_A τερματίσει, **αποδέχεται**. Αλλιώς, πηγαίνει στο 4..
4. Στην 2η ταινία εκτελεί M_B για x .
5. Αν η M_B τερματίσει, **αποδέχεται**. Αλλιώς, **απορρίπτει**.” $\square$

■ Η M χρειάζεται υπολογιστικό χρόνο: $O(n^k) + O(n^\ell) = O(n^m)$, $m = \max\{k, \ell\}$.

■ Επομένως, αν κατασκευάσουμε μια TM 1-ταινίας για την $A \cup B$ θα χρειαστούμε $O(n^{2m})$ υπολογιστικό χρόνο.

Κλάση P

Θεώρημα. Αν A μια διαγνώσιμη σε πολυωνυμικό χρόνο γλώσσα, τότε και η γλώσσα $\bar{A}$ είναι διαγνώσιμη σε πολυωνυμικό χρόνο. [$A \in P \Rightarrow \bar{A} \in P$]

Απόδειξη. Έστω ότι η A διαγιγνώσκεται από την ντετερμινιστική ΤΜ M σε χρόνο $O(n^m)$, για κάποιο $m \in \mathbb{N}$.

Κατασκευάζουμε μια ντετερμινιστική ΤΜ M' η οποία να προσομοιώνει την M και να αποδέχεται τις εισόδους τις οποίες απορρίπτει η M σε χρόνο $O(n^m)$. Προφανώς, η M' αποδέχεται την $\bar{A}$.

Άρα $\bar{A} \in TIME(n^m)$. □

Κλάση P

■ Η τάξη πολυπλοκότητας co-P περιέχει όλες τις γλώσσες των οποίων το συμπλήρωμα είναι στην $\mathbf{P}$.

$$\text{co-P} = \{L \mid \bar{L} \in \mathbf{P}\}.$$

■ Ισχύει $\text{co-P} = \mathbf{P}$;

$\text{co-P} \subseteq \mathbf{P}$: Έστω $L \in \mathbf{P}$. Εφόσον, $\mathbf{P}$ κλειστή ως προς το συμπλήρωμα τότε $\bar{L} \in \mathbf{P}$. Συνεπώς, $L = \bar{\bar{L}} \in \text{co-P}$.

$\mathbf{P} \subseteq \text{co-P}$: Έστω $\bar{L} \in \text{co-P}$. Εφόσον, $\mathbf{P}$ κλειστή ως προς το συμπλήρωμα τότε $L \in \mathbf{P}$. Άρα, $L \in \text{co-P}$. □

Κλάση P

■ Modular exponentiation, $a^b \equiv c(\text{mod } p)$, π.χ. $243 = 3^5 (= 3^4 \cdot 3 = 4 \cdot 3) \equiv 5(\text{mod } 7)$.

■ Ορίζουμε την κλάση προβλημάτων *MODEXP* (modular exponentiation), ως εξής,

$$\text{MODEXP} = \{ \langle a, b, c, p \rangle \mid a, b, c, \text{ και } p \text{ είναι δυαδικοί ακέραιοι τ.ω. } a^b \equiv c(\text{mod } p) \}.$$

■ Θα δείξουμε ότι $\text{MODEXP} \in \mathbf{P}$.

Επειδή το b είναι δυαδικός ακέραιος, έστω $b = b_1 b_2 \dots b_{n-1} b_n$. Άρα η δεκαδική αναπαράσταση του b είναι $\sum_{k=1}^n b_{n-k+1} 2^{k-1}$.

Παρατηρούμε ότι, $(10)_2 = 2^1$, δηλαδή $a^{(10)_2} = a^2$ και $(1000)_2 = 2^3$, δηλαδή $a^{(1000)_2} = (a^2)^3 = ((a^2)^2)^2$, κτλ.

Άρα, $a^b = a^{\sum_{k=1}^n b_{n-k+1} 2^{k-1}} = a^{b_1 2^{n-1}} a^{b_2 2^{n-2}} \dots a^{b_n 2^0}$.

...

Κλάση P

Απόδειξη. Έχουμε ότι $a^b = a^{\sum_{k=1}^n b_{n-k+1}2^{k-1}} = a^{b_12^{n-1}} a^{b_22^{n-2}} \dots a^{b_n2^0}$.

Ιδέα. Προχωράμε bit-bit. Κάθε φορά τετραγωνίζουμε το αποτέλεσμα, και περνάμε στην επόμενη δύναμη του 2. Αν το τρέχον bit είναι 1, τότε κάνουμε επιπλέον ένα πολλαπλασιασμό με το a .

Κατασκευάζουμε την εξής TM για να διαγνώσουμε το $MODEXP$,

$M =$ “Για είσοδο $\langle a, b, c, p \rangle$, όπου a, b, c, p δυαδικοί ακέραιοι :

1. Θέτει $T = 1$ και $(n = \lceil \log_2 b \rceil)$.
2. Για $i = 1, \dots, n$:
3. Αν $b_i = 1$, τότε $T = (a \cdot (T^2) \pmod p)$.
4. Αν $b_i = 0$, τότε $T = (T^2) \pmod p$.
5. Επιστρέφει $T \pmod p$.
6. Ελέγχει αν $T = c \pmod p$. Αν ναι, **αποδέχεται**, αλλιώς **απορρίπτει**.”

Κλάση P

...

Υποθέτουμε ότι το a, b, c, p έχουν το πολύ m bits (άρα $n \leq m$).

Δύο m -bit αριθμοί κοστίζουν $O(m^2)$ χρόνο για πολλαπλασιασμό και διαίρεση. Άρα και για υπολογισμό υπολοίπου.

Συνεπώς, κάθε βήμα i κοστίζει $O(m^2)$ χρόνο. Το συνολικό κόστος του 2. είναι $O(m^2) \times \log m = O(m^2 \log m)$ το οποίο είναι το κυρίαρχο κόστος όλων των βημάτων.

Άρα, η χρονική πολυπλοκότητα της TM είναι πολυωνυμική στο μήκος της εισόδου. □

Κλάση P

■ Ορίζουμε την κλάση προβλημάτων $\Sigma\Gamma\mathcal{N}\Theta\mathcal{E}\mathcal{T}\mathcal{O}\Sigma$, ως εξής

$$\Sigma\Gamma\mathcal{N}\Theta\mathcal{E}\mathcal{T}\mathcal{O}\Sigma := \{p \mid p \text{ σύνθετος ακέραιος αριθμός}\}$$

■ Να δείξτε ότι $\Sigma\Gamma\mathcal{N}\Theta\mathcal{E}\mathcal{T}\mathcal{O}\Sigma \in \mathbf{P}$.

Κατασκευάζουμε την εξής TM:

$M =$ “Για είσοδο $\langle p, q \rangle$, όπου p, q ακέραιοι αριθμοί:

1. Ελέγχει αν $1 < q < p$. Αν όχι, **απορρίπτει**.
2. Ελέγχει αν q διαιρεί το p . Αν ναι, **αποδέχεται**, αλλιώς **απορρίπτει**.”

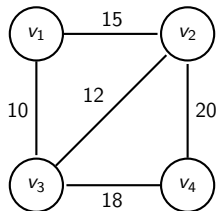
(Ορθότητα) Ο p είναι σύνθετος ακέραιος αριθμός $\iff$ υπάρχει αριθμός $q \in (1, p)$ τ.ω. να διαιρεί το p . $\square$

Αλγόριθμος Kruskal

- Δοθέντος ενός γράφου G βρείτε το ελάχιστο παραγόμενο δέντρο [MWST minimum-weight spanning tree]
- Ο αλγοριθμος του Kruskal εξετάζει τις ακμές σε αύξουσα σειρά βάρους.
- Σε κάθε βήμα του αλγορίθμου, εξετάζεται η ακμή με το μικρότερο βάρος (που δεν έχει ήδη εξεταστεί) και προστίθεται στο MWST εφόσον, δε δημιουργεί κύκλο με τις ακμές που έχουν ήδη προστεθεί. [cut property]
- Η διαδικασία ολοκληρώνεται όταν προστεθούν $|V| - 1$ ακμές. [V πλήθος κορυφών]

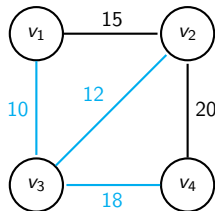
Αλγόριθμος Kruskal

■ Δοθέντος ενός γράφου G βρείτε το ελάχιστο παραγόμενο δέντρο MWST [minimum-weight spanning tree]



Αλγόριθμος Kruskal

■ Δοθέντος ενός γράφου G βρείτε το ελάχιστο παραγόμενο δέντρο [MWST minimum-weight spanning tree]



■ Ο Αλγόριθμος Kruskal βρίσκει το MWST [είναι ένας άπληστος αλγόριθμος]

Κλάση P

■ $\text{MWST} \in \text{P}$.

Κατασκευάζουμε TM M η οποία να διαγιγνώσκει για το MWST εκτελώντας τις παρακάτω ενέργειες σε είσοδο $\langle G \rangle$, όπου ο G έχει V ακμές

$M =$ “Για είσοδο $\langle G \rangle$:

1. Ακμές σε αύξουσα σειρά ως προς το βάρος τους.
2. Ξεκινάει από $T = \emptyset$.
3. Προσθέτει μια ακμή στο T εκτός αν δημιουργεί κύκλο.
Αποδέχεται αν καλύψει όλες τις κορυφές.
4. Αλλιώς, απορρίπτει.” $\square$

■ Η TM M παράγει ένα MWST.

■ Χρειάζεται $O(|V|^2)$, άρα τρέχει σε πολυωνυμικό χρόνο.

Κλάση P

Μερικά σχόλια

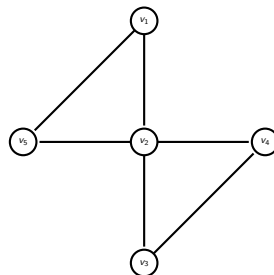
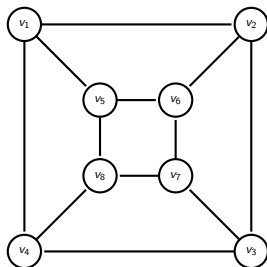
1. Οι greedy algorithms ταξινομούν τα στοιχεία και στη συνέχεια τα επεξεργάζονται είτε επιλέγοντάς τα είτε αγνοώντας τα, χωρίς ποτέ να αλλάζουν μια απόφαση.

Η απόδειξη της ορθότητας γίνεται με επαγωγή στον αριθμό των βημάτων.

2. Δυναμικός προγραμματισμός. Προϋπολογισμός μερικών λύσεων ξεκινώντας από τις βασικές περιπτώσεις.

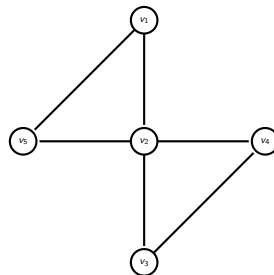
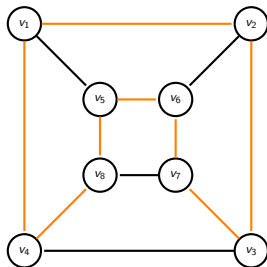
Ο χρόνος εκτέλεσης μπορεί να μην είναι πολυωνυμικός.

Μονοπάτι Hamilton

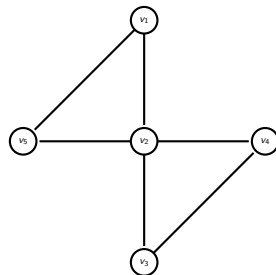
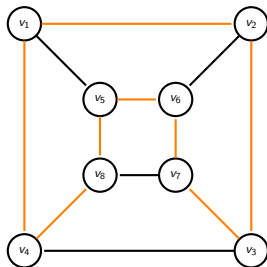


■ Μερικές εφαρμογές: Βιοπληροφορική, κατασκευή κυκλωμάτων, ορισμός δικτύων, μεταφορές, logistics, Μοριακή Βιολογία, μετάδοση δεδομένων, κτλ

Μονοπάτι Hamilton



Μονοπάτι Hamilton



■ Ορίζουμε την κλάση προβλημάτων *HAMILTON*,

$$HAMPATH := \left\{ \langle G, s, t \rangle \mid \begin{array}{l} \langle G, s, t \rangle \in PATH \text{ και το μονοπάτι} \\ \text{περνάει από κάθε κορυφή του } G \\ \text{μια φορά.} \end{array} \right\}.$$

■ Αν $s = t$ τότε ο γράφος είναι Hamiltonian.

Μονοπάτι Hamilton

■ Ισχύει $HAMPATH \in P$;

Κατασκευάζουμε ΤΜ M η οποία να διαγιγνώσκει για το $HAMPATH$ εκτελώντας τις παρακάτω ενέργειες σε είσοδο $\langle G, s, t \rangle$, όπου ο G έχει m κορυφές

$M =$ “Για είσοδο $\langle G, s, t \rangle$:

1. Για κάθε μονοπάτι μήκους m κορυφών στον G ελέγχει αν είναι μονοπάτι Hamilton. από το s στο t , **αποδέχεται** αν υπάρχει.
2. **Απορρίπτει** αν δεν βρεθεί κανένα τέτοιο μονοπάτι.” $\square$

■ Αν $m! > 2^m$ πλήθος μονοπατιών μήκους m , τότε ο αλγόριθμος τρέχει σε εκθετικό χρόνο και όχι σε πολυωνυμικό!

■. Επομένως, τι συμπέρασμα βγάζουμε για την ερώτηση: $HAMPATH \in P$;

Μονοπάτι Hamilton

■ Ισχύει $HAMPATH \in P$;

Κατασκευάζουμε ΤΜ M η οποία να διαγιγνώσκει για το $HAMPATH$ εκτελώντας τις παρακάτω ενέργειες σε είσοδο $\langle G, s, t \rangle$, όπου ο G έχει m κορυφές

$M =$ “Για είσοδο $\langle G, s, t \rangle$:

1. Για κάθε μονοπάτι μήκους m κορυφών στον G ελέγχει αν είναι μονοπάτι Hamilton. από το s στο t , **αποδέχεται** αν υπάρχει.
2. **Απορρίπτει** αν δεν βρεθεί κανένα τέτοιο μονοπάτι.” $\square$

■ Αν $m! > 2^m$ πλήθος μονοπατιών μήκους m , τότε ο αλγόριθμος τρέχει σε εκθετικό χρόνο και όχι σε πολυωνυμικό!

■. Επομένως, τι συμπέρασμα βγάζουμε για την ερώτηση: $HAMPATH \in P$; Άλυτο πρόβλημα, συνδέεται με την παραγοντοποίηση.