

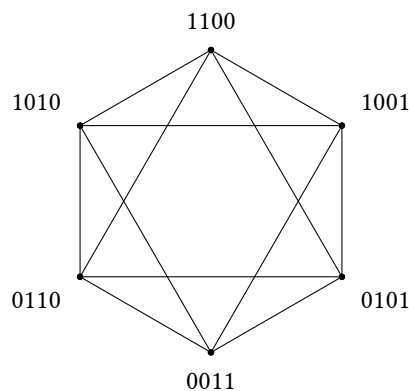
Καθηγητής Π. Λουρίδας

Τμήμα Διοικητικής Επιστήμης και Τεχνολογίας

Οικονομικό Πανεπιστήμιο Αθηνών

## Γράφοι Μεταθέσεων

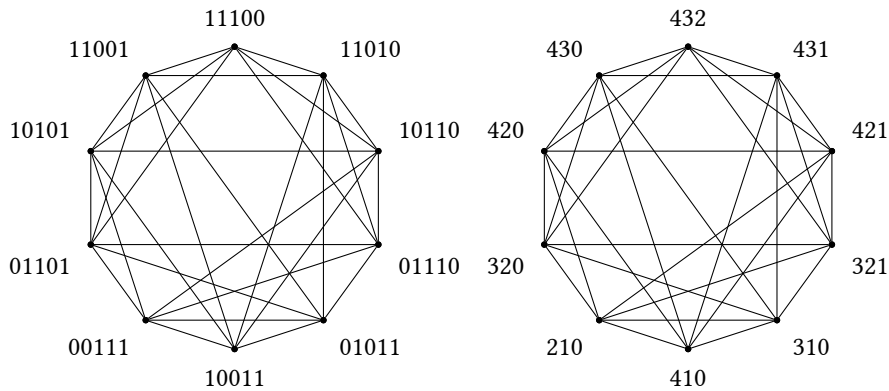
Ένας γράφος μεταθέσεων (transposition graph) είναι ένας γράφος που οι κόμβοι του αντιστοιχούν στις διαφορετικές μεταθέσεις (permutations) ενός συνόλου συμβόλων και οι σύνδεσμοι είναι μεταξύ κόμβων που διαφέρουν κατά μόνο μια μετάθεση. Αν έχουμε δύο ειδών σύμβολα, όπως 0 και 1, τότε οι κόμβοι αντιστοιχούν σε ακολουθίες 0 και 1. Για παράδειγμα, αν έχουμε δύο 0 και δύο 1, τότε ο γράφος μεταθέσεων είναι ο παρακάτω:



Αν έχουμε δύο ειδών σύμβολα, έστω  $s$  από 0 και  $t$  από 1, τότε οι κόμβοι του γράφου μεταθέσεων είναι  $\binom{s+t}{s} = \binom{s+t}{t}$ . Αν θέσουμε  $n = s + t$ , ο κάθε κόμβος του γράφου συνδέεται με  $\frac{1}{2}[n^2 - (s^2 + t^2)]$  κόμβους.

Αφού ο γράφος μεταθέσεων είναι ένας πλήρως συνδεδεμένος γράφος, μπορούμε από οποιονδήποτε κόμβο του να επισκεφτούμε κάθε άλλο κόμβο του γράφου, διαγράφοντας ένα μονοπάτι από κόμβο σε κόμβο. Κάθε τέτοιο μονοπάτι είναι ένας τρόπος να απαριθμήσουμε τις μεταθέσεις του γράφου. Εμάς όμως εδώ μας ενδιαφέρουν τα μονοπάτια που θα πληρούν δύο συγκεκριμένες ιδιότητες.

Για να δούμε ποιες είναι οι ιδιότητες αυτές, πρέπει πρώτα να δούμε έναν τρόπο περιγραφής των κόμβων αυτών των γράφων μεταθέσεων. Αφού κάθε κόμβος του γράφου μεταθέσεων που προκύπτει από δύο σύμβολα είναι μία δυαδική συμβολοσειρά, μπορούμε να περιγράψουμε κάθε κόμβο με τις θέσεις όπου εμφανίζονται τα 1. Θα ονομάσουμε αυτή την αναπαράσταση *αναπαράσταση δεικτών* (indices representation), επειδή χρησιμοποιεί τους δείκτες των μονάδων. Έτσι, οι κόμβοι του γράφου μεταθέσεων για  $s = 2, t = 3$  μπορούν να γραφούν είτε όπως παρακάτω στα αριστερά, είτε όπως παρακάτω στα δεξιά:



Ένα μονοπάτι στο οποίο οι κόμβοι εμφανίζονται ώστε κάθε πρόθεμα να εμφανίζεται σε διαδοχικούς κόμβους είναι μια *ακολουθία genlex* (genlex sequence, από το generalised lexicographic sequence). Ένα τέτοιο μονοπάτι είναι το εξής:

$$432 \rightarrow 431 \rightarrow 430 \rightarrow 420 \rightarrow 421 \rightarrow 410 \rightarrow 310 \rightarrow 320 \rightarrow 321 \rightarrow 210$$

Μια ακολουθία ονομάζεται *ομογενής* (homogeneous) αν όταν μετακινούμαστε από ένα στοιχείο της στο επόμενο αλλάζει μόνο ένα ψηφίο. Η παραπάνω ακολουθία είναι genlex, αλλά δεν είναι ομογενής. Η παρακάτω είναι ομογενής genlex. Παρατηρούμε ότι μια ακολουθία genlex δεν είναι απαραίτητα ταξινομημένη.

$$432 \rightarrow 431 \rightarrow 430 \rightarrow 410 \rightarrow 420 \rightarrow 421 \rightarrow 321 \rightarrow 320 \rightarrow 310 \rightarrow 210$$

Επιπλέον, μια ακολουθία μπορεί να είναι ομογενής χωρίς να είναι genlex:

$$420 \rightarrow 410 \rightarrow 430 \rightarrow 432 \rightarrow 431 \rightarrow 421 \rightarrow 321 \rightarrow 320 \rightarrow 310 \rightarrow 210$$

Οι δύο ιδιότητες που θέλουμε να πληρούν τα μονοπάτια μας είναι ακριβώς να είναι ταυτόχρονα ομογενείς και genlex ακολουθίες.

Για να βρούμε τέτοια μονοπάτια μπορούμε να εργαστούμε με δύο τρόπους. Ο πρώτος τρόπος είναι να κάνουμε κατά βάθος διάσχιση του γράφου. Αν κατά τη διάρκεια της διάσχισης κρατάμε σε μία στίβα τους κόμβους που έχουμε επισκεφτεί, όταν το μέγεθος της στίβας είναι ίσο με το πλήθος των κόμβων του γράφου, οι κόμβοι στη στίβα αποτελούν ένα μονοπάτι που επισκέπτεται όλους τους κόμβους του γράφου.

Στην κατά βάθος διάσχιση του γράφου ελέγχουμε κάθε γείτονα του κόμβου που επισκεπτόμαστε και αν δεν έχουμε επισκεφτεί τον γείτονα τότε συνεχίζουμε αναδρομικά τη διάσχιση από τον εν λόγω γείτονα. Για να βρούμε μονοπάτια που να είναι ομογενείς genlex ακολουθίες θα πρέπει να ελέγχουμε δύο επιπλέον συνθήκες. Πρώτον, ότι ο κόμβος που θα επισκεφτούμε διαφέρει από τον κόμβο στον οποίο βρισκόμαστε

κατά μόνο ένα ψηφίο στην αναπαράσταση δεικτών. Αυτό εξασφαλίζει ότι δεν παραβιάζεται η απαίτηση της ομογένειας. Δεύτερον, ότι ο κόμβος που θα επισκεφτούμε, αν τον προσθέσουμε στο μονοπάτι που κατασκευάζουμε, το μονοπάτι θα είναι ακολouthία genlex.

Ο δεύτερος τρόπος για τον εντοπισμό τέτοιων μονοπατιών στηρίζεται στη χρήση του ισορροπημένου τριαδικού συστήματος αρίθμησης (balanced ternary system) το οποίο έχει βάση το τρία και τα ψηφία έχουν τις τιμές -1, 0, και 1. Για να γράψουμε αριθμούς στο ισορροπημένο τριαδικό σύστημα χρησιμοποιούμε τα σύμβολα -, 0, και +. Για παράδειγμα, το ++0-0- αντιστοιχεί στον αριθμό 314:

$$-1 \times 3^0 + 0 \times 3^1 + (-1) \times 3^2 + 0 \times 3^3 + 1 \times 3^4 + 1 \times 3^5 = 314$$

Αν έχουμε μια συμβολοσειρά στο ισορροπημένο τριαδικό σύστημα αρίθμησης, μπορούμε να ορίσουμε ως *R-block* ένα μέρος της συμβολοσειράς με μορφή  $-^{k+1}$ ,  $k \geq 0$  του οποίου προηγείται ένα 0 και δεν ακολουθεί ένα -. Ένα *L-block* είναι ένα μέρος της συμβολοσειράς με μορφή  $+^k$ ,  $k \geq 0$ , που ακολουθεί 0. Στη συμβολοσειρά  $\boxed{+00++-++}\boxed{+000}\boxed{-}$  έχουμε από τα αριστερά προς τα δεξιά ένα L-block, άλλο ένα L-block, και ένα R-block. Τα blocks δεν μπορούν να επικαλύπτουν το ένα το άλλο.

Αν έχουμε μια συμβολοσειρά η οποία περιέχει τουλάχιστον ένα block, τότε μπορούμε να δημιουργήσουμε το διάδοχό της ως εξής. Αν το δεξιότερο block είναι ένα R-block, τότε αντικαθιστούμε το δεξιότερο  $0^{-k+1}$  με  $-+^k0$ . Διαφορετικά, αντικαθιστούμε το δεξιότερο  $+^k0$  με  $0+^{k+1}$ . Επίσης, αλλάζουμε το πρώτο πρόσημο, αν υπάρχει, που ακολουθεί το block που μόλις αλλάξαμε. Μπορούμε να δούμε τη διαδικασία στην παρακάτω ακολουθία:

$000\boxed{+} \rightarrow 00\boxed{-}0 \rightarrow 00\boxed{-}0 \rightarrow 0\boxed{+}00 \rightarrow 0\boxed{-}0\boxed{+}0 \rightarrow 0\boxed{-}00\boxed{+} \rightarrow -000\boxed{-} \rightarrow -00\boxed{-}0 \rightarrow -0\boxed{-}00 \rightarrow --000$

Ή σε αυτήν:

$\boxed{+}000\boxed{-} \rightarrow \boxed{+}00\boxed{-}0 \rightarrow \boxed{+}0\boxed{-}00 \rightarrow \boxed{+}-000 \rightarrow 0\boxed{+}00 \rightarrow 0\boxed{+}0\boxed{+}0 \rightarrow 0\boxed{+}00\boxed{+} \rightarrow 00\boxed{+}0\boxed{-} \rightarrow 00\boxed{+}-0 \rightarrow 000\boxed{++}$

Αν τώρα αντικαταστήσουμε τα σύμβολα του ισορροπημένου τριαδικού συστήματος με τους κανόνες:  $+ \rightarrow 0$ ,  $- \rightarrow 0$ ,  $0 \rightarrow 1$ , τότε παίρνουμε τις εξής ακολουθίες:

$11100 \rightarrow 11010 \rightarrow 11001 \rightarrow 10011 \rightarrow 10101 \rightarrow 10110 \rightarrow 01110 \rightarrow 01101 \rightarrow 01011 \rightarrow 00111$

και

$01110 \rightarrow 01101 \rightarrow 01011 \rightarrow 00111 \rightarrow 10011 \rightarrow 10101 \rightarrow 10110 \rightarrow 11010 \rightarrow 11001 \rightarrow 11100$

ή ισοδύναμα αν χρησιμοποιήσουμε αναπαράσταση δεικτών:

$432 \rightarrow 431 \rightarrow 430 \rightarrow 410 \rightarrow 420 \rightarrow 421 \rightarrow 321 \rightarrow 320 \rightarrow 310 \rightarrow 210$

και

$$321 \rightarrow 320 \rightarrow 310 \rightarrow 210 \rightarrow 410 \rightarrow 420 \rightarrow 421 \rightarrow 431 \rightarrow 430 \rightarrow 432$$

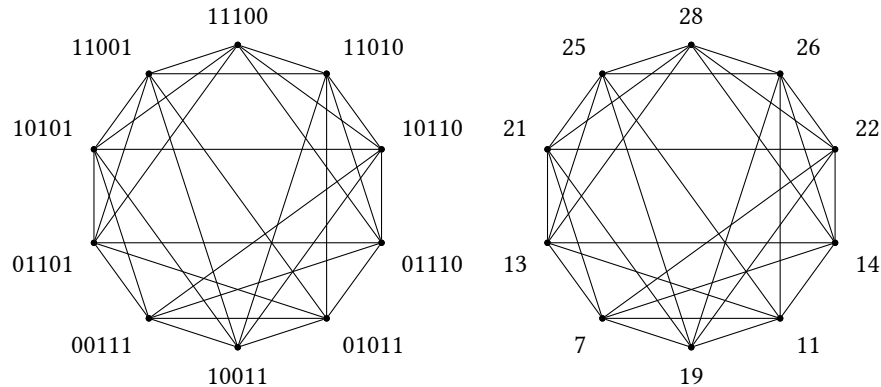
Μπορείτε να επιβεβαιώσετε ότι και οι δύο ακολουθίες είναι ομογενείς genlex ακολουθίες στον γράφο μεταθέσεων  $s = 2, t = 3$ !

Αυτό δεν είναι σύμπτωση. Ας ορίσουμε με  $\sigma_j$ , όπου  $0 \leq j < 2^{s-1}$ , τη συμβολοσειρά  $0^t - \tau_j$  όπου το  $\tau_j$  είναι η συμβολοσειρά μήκους  $s - 1$  της οποίας ο χαρακτήρας στη θέση  $k$  είναι το  $-$  αν ο  $k$  χαρακτήρας της δυαδικής αναπαράστασης του  $j$  είναι 1, διαφορετικά είναι το  $+$ . Αν  $s = 2, t = 3$ , έχουμε  $\sigma_0 = 000-+, \sigma_1 = 000--$ . Αν  $s = 3, t = 3$ , έχουμε  $\sigma_0 = 000-++, \sigma_1 = 000-+-, \sigma_2 = 000+--, \sigma_3 = 000+++$ . Γενικά,  $\sigma_0 = 0^t-++ \dots +, \sigma_1 = 0^t--+ \dots +, \dots, \sigma_{2^{s-1}-1} = 0^t--- \dots -$ .

Αν ξεκινήσουμε από ένα  $\sigma_i$  και εφαρμόσουμε τους κανόνες αντικατάστασης που περιγράψαμε παραπάνω, θα πάρουμε πάντοτε ένα μονοπάτι που είναι μια ομογενής genlex ακολουθία.

Σκοπός της εργασίας είναι η συγγραφή προγράμματος που βρίσκει μονοπάτια που διασχίζουν πλήρως έναν γράφο μεταθέσεων, και τα οποία αντιστοιχούν σε ομογενείς genlex ακολουθίες. Η εύρεση των μονοπατιών θα γίνεται με δύο τρόπους, με κατά βάθος διάσχιση του γράφου και δημιουργώντας αλυσίδες αντικατάστασης από τα  $\sigma_j, 0 \leq j < 2^{s-1}$ , που προκύπτουν για τον εν λόγω γράφο.

Όσον αφορά την αναπαράσταση του γράφου μεταθέσεων, δεδομένου ότι οι κόμβοι του είναι αριθμοί στο δυαδικό σύστημα αρίθμησης, στο πρόγραμμά σας οι κόμβοι θα είναι οι αντίστοιχοι αριθμοί στο δεκαδικό σύστημα αρίθμησης:



Αυτό σημαίνει ότι για να δημιουργήσετε το γράφο θα πρέπει να χρησιμοποιήσετε δυαδικούς τελεστές όπως XOR ( $\wedge$ ), ολίσθηση αριστερά ( $\ll$ ), OR ( $\vee$ ), κ.λπ.

## Απαιτήσεις Προγράμματος

Κάθε φοιτητής θα εργαστεί σε αποθετήριο στο GitHub. Για να αξιολογηθεί μια εργασία θα πρέπει να πληροί τις παρακάτω προϋποθέσεις:

- Για την υποβολή της εργασίας θα χρησιμοποιηθεί το ιδιωτικό αποθετήριο του φοιτητή που δημιουργήθηκε για τις ανάγκες του μαθήματος και του έχει αποδοθεί. Το αποθετήριο αυτό έχει όνομα του τύπου `username-algo-assignments`, όπου `username` είναι το όνομα του φοιτητή στο GitHub. Για παράδειγμα, το σχετικό αποθετήριο του διδάσκοντα θα ονομαζόταν `louridas-algo-assignments` και θα ήταν προσβάσιμο στο <https://github.com/dmst-algorithms-course/louridas-algo-assignments>. Τυχόν άλλα αποθετήρια απλώς θα αγνοηθούν.
- Μέσα στο αποθετήριο αυτό θα πρέπει να δημιουργηθεί ένας κατάλογος `assignment-2025-1`.
- Μέσα στον παραπάνω κατάλογο το πρόγραμμα θα πρέπει να αποθηκευτεί με το όνομα `transposition_graphs.py`.
- Δεν επιτρέπεται η χρήση έτοιμων βιβλιοθηκών γράφων ή τυχόν έτοιμων υλοποιήσεων των αλγορίθμων, ή τμημάτων αυτών, εκτός αν αναφέρεται ρητά ότι επιτρέπεται.
- Επιτρέπεται η χρήση δομών δεδομένων της Python όπως στοίβες, λεξικά, σύνολα, κ.λπ.
- Επιτρέπεται η χρήση των παρακάτω βιβλιοθηκών ή τμημάτων τους όπως ορίζεται:
  - `sys.argv`
  - `argparse`
- Το πρόγραμμα θα πρέπει να είναι γραμμένο σε Python 3.
- Η εργασία είναι αποκλειστικά ατομική. Δεν επιτρέπεται συνεργασία μεταξύ φοιτητών στην εκπόνησή της, με ποινή το μηδενισμό. Επιπλέον η εργασία δεν μπορεί να είναι αποτέλεσμα συστημάτων Τεχνητής Νοημοσύνης (όπως ChatGPT). Ειδικότερα όσον αφορά το τελευταίο σημείο προσέξτε ότι τα συστήματα αυτά χωλαίνουν στην αλγοριθμική σχέση, άρα τυχόν προτάσεις που κάνουν σε σχετικά θέματα μπορεί να είναι λανθασμένες. Επιπλέον, αν θέλετε να χρησιμοποιήσετε τέτοια συστήματα, τότε αν και κάποιος άλλος το κάνει αυτό, μπορεί οι προτάσεις που θα λάβετε να είναι παρόμοιες, οπότε οι εργασίες θα παρουσιάσουν ομοιότητες και άρα θα μηδενιστούν.
- Η έξοδος του προγράμματος θα πρέπει να περιλαμβάνει μόνο ό,τι φαίνεται στα παραδείγματα που παρατίθενται. *Η φλυαρία δεν επιβραβεύεται.*

## Χρήση του GitHub

Όπως αναφέρθηκε το πρόγραμμά σας για να αξιολογηθεί θα πρέπει να αποθηκευθεί στο GitHub. Επιπλέον, θα πρέπει το GitHub να χρησιμοποιηθεί καθόλη τη διάρκεια της ανάπτυξης του.

Αυτό σημαίνει ότι *δεν θα ανεβάσετε στο GitHub απλώς την τελική λύση του προβλήματος μέσω της λειτουργίας "Upload files"*. Στο GitHub θα πρέπει να φαίνεται το ιστορικό

της συγγραφής του προγράμματος. Αυτό συνάδει και με τη φιλοσοφία του εργαλείου: λέμε “commit early, commit often”. Εργαζόμαστε σε ένα πρόγραμμα και κάθε μέρα, ή όποια στιγμή έχουμε κάνει κάποιο σημαντικό βήμα, αποθηκεύουμε την αλλαγή στο GitHub. Αυτό έχει σειρά ευεργετικών αποτελεσμάτων:

- Έχουμε πάντα ένα αξιόπιστο εφεδρικό μέσο στο οποίο μπορούμε να ανατρέξουμε αν κάτι πάει στραβά στον υπολογιστή μας (μας γλιτώνει από πανικούς του τύπου: ένα βράδυ πριν από την παράδοση ο υπολογιστής μας ή ο δίσκος του πνέει τα λούστια, και εμείς τι θα υποβάλουμε στον Λουρίδα;).
- Καθώς έχουμε πλήρες ιστορικό των σημαντικών αλλαγών και εκδόσεων του προγράμματός μας, μπορούμε να επιστρέψουμε σε μία προηγούμενη αν συνειδητοποιήσουμε κάποια στιγμή ότι πήραμε λάθος δρόμο (μας γλιτώνει από πανικούς του τύπου: μα το πρωί δούλευε σωστά, τι στο καλό έκανα και τώρα δεν παίζει τίποτε;).
- Καθώς φαίνεται η πρόοδος μας στο GitHub, επιβεβαιώνουμε ότι το πρόγραμμα δεν είναι αποτέλεσμα της όποιας επιφοίτησης (μας γλιτώνει από πανικούς του τύπου: καλά, πώς το έλυσες το πρόβλημα αυτό με τη μία, μόνος σου;).
- Αν δουλεύετε σε μία ομάδα που βεβαίως δεν είναι καθόλου η περίπτωση μας εδώ, μπορούν όλοι να εργάζονται στα ίδια αρχεία στο GitHub εξασφαλίζοντας ότι ο ένας δεν γράφει πάνω στις αλλαγές του άλλου. Παρά το ότι η εργασία αυτή είναι ατομική, καλό είναι να αποκτάτε τριβή με το εργαλείο git και την υπηρεσία GitHub μιας και χρησιμοποιούνται ευρέως και όχι μόνο στη συγγραφή κώδικα.

Άρα επενδύστε λίγο χρόνο στην εκμάθηση των git / GitHub, των οποίων ο σωστός τρόπος χρήσης είναι μέσω γραμμής εντολών (command line), ή ειδικών προγραμμάτων (clients) ή μέσω ενοποίησης στο περιβάλλον ανάπτυξης (editor, IDE).

### Τελικό Πρόγραμμα

Το πρόγραμμα θα καλείται ως εξής (όπου python η κατάλληλη εντολή στο εκάστοτε σύστημα):

```
python transposition_graphs.py s t {graph,dfs,bfs} [start]
```

Η σημασία των παραμέτρων του προγράμματος είναι:

- s: το πλήθος των συμβόλων 0.
- t το πλήθος των συμβόλων 1.
- graph: αν δοθεί, το πρόγραμμα θα εμφανίσει στην οθόνη τον γράφο μεταθέσεων και θα σταματήσει.
- dfs: αν δοθεί, το πρόγραμμα θα εμφανίσει στην οθόνη τα μονοπάτια που θα προκύψουν από την κατά βάθος διάσχιση του γράφου, είτε από το σύνολο των κόμβων του γράφου είτε αναλόγως την παράμετρο start.

- `bts` αν δοθεί, το πρόγραμμα θα εμφανίσει στην οθόνη τα μονοπάτια που θα προκύψουν με τη μέθοδο του ισορροπημένου τριαδικού συστήματος αρίθμησης.
- `start` αν δοθεί και έχει δοθεί η παράμετρος `dfs`, προσδιορίζει τον κόμβο εκκίνησης για την κατά βάθος αναζήτηση.

## Παραδείγματα

### Παράδειγμα 1

Αν ο χρήστης του προγράμματος δώσει:

```
python transposition_graphs.py 2 4 graph
```

το πρόγραμμά σας θα πρέπει να εμφανίσει ακριβώς τα παρακάτω:

```
60 -> [58, 57, 54, 53, 46, 45, 30, 29]
58 -> [60, 57, 51, 54, 43, 46, 27, 30]
57 -> [60, 58, 53, 51, 45, 43, 29, 27]
54 -> [60, 58, 51, 53, 39, 46, 23, 30]
53 -> [60, 57, 54, 51, 45, 39, 29, 23]
46 -> [60, 58, 54, 43, 39, 45, 15, 30]
45 -> [60, 57, 53, 46, 43, 39, 29, 15]
30 -> [60, 58, 54, 46, 27, 23, 15, 29]
29 -> [60, 57, 53, 45, 30, 27, 23, 15]
51 -> [58, 57, 54, 53, 43, 39, 27, 23]
43 -> [58, 57, 46, 45, 51, 39, 27, 15]
27 -> [58, 57, 30, 29, 51, 43, 23, 15]
39 -> [54, 53, 46, 45, 51, 43, 23, 15]
23 -> [54, 53, 30, 29, 51, 27, 39, 15]
15 -> [46, 45, 30, 29, 43, 27, 39, 23]
```

### Παράδειγμα 2

Αν ο χρήστης του προγράμματος δώσει:

```
python transposition_graphs.py 2 4 bts
```

το πρόγραμμά σας θα πρέπει να εμφανίσει ακριβώς τα παρακάτω:

```
['0000-+', '000-0-', '000--0', '00-+00', '00-0+0', '00-00+',
↳ '0-000-', '0-00-0', '0-0-00', '0--000', '-+0000', '-0+000',
↳ '-00+00', '-000+0', '-0000+']
[111100, 111010, 111001, 110011, 110101, 110110, 101110, 101101,
↳ 101011, 100111, 001111, 010111, 011011, 011101, 011110]
[5432, 5431, 5430, 5410, 5420, 5421, 5321, 5320, 5310, 5210,
↳ 3210, 4210, 4310, 4320, 4321]
[60, 58, 57, 51, 53, 54, 46, 45, 43, 39, 15, 23, 27, 29, 30]
```

```
['0000--', '000-+0', '000-0+', '00-00-', '00-0-0', '00--00',
↳ '0-+000', '0-0+00', '0-00+0', '0-000+', '-0000-', '-000-0',
↳ '-00-00', '-0-000', '--0000']
[111100, 111001, 111010, 110110, 110101, 110011, 100111, 101011,
↳ 101101, 101110, 011110, 011101, 011011, 010111, 001111]
[60, 57, 58, 54, 53, 51, 39, 43, 45, 46, 30, 29, 27, 23, 15]
```

Στη συγκεκριμένη περίπτωση, το πρόγραμμα βρήκε τα δύο μονοπάτια που είναι ομογενείς genlex ακολουθίες ξεκινώντας από τον κόμβο 111100 του γράφου. Κάθε μονοπάτι εμφανίζεται με τέσσερις τρόπους: χρησιμοποιώντας ισορροπημένο τριαδικό σύστημα, δυαδικό σύστημα, αναπαράσταση δεικτών, και ακέραιους αριθμούς. Η κάθε αναπαράσταση δίνεται σε μία γραμμή (επειδή στο χαρτί δεν χωράει, το σύμβολο  $\hookrightarrow$  υποδεικνύει απλώς ότι η γραμμή συνεχίζεται).

### Παράδειγμα 3

Αν ο χρήστης του προγράμματος δώσει:

```
python transposition_graphs.py 2 4 dfs
```

το πρόγραμμά σας θα πρέπει να εμφανίσει ακριβώς τα παρακάτω:

```
111100, 111010, 111001, 110011, 110101, 110110, 101110, 101101,
↳ 101011, 100111, 001111, 010111, 011011, 011101, 011110]
[5432, 5431, 5430, 5410, 5420, 5421, 5321, 5320, 5310, 5210,
↳ 3210, 4210, 4310, 4320, 4321]
[60, 58, 57, 51, 53, 54, 46, 45, 43, 39, 15, 23, 27, 29, 30]
[111100, 111001, 111010, 110110, 110101, 110011, 100111, 101011,
↳ 101101, 101110, 011110, 011101, 011011, 010111, 001111]
[5432, 5430, 5431, 5421, 5420, 5410, 5210, 5310, 5320, 5321,
↳ 4321, 4320, 4310, 4210, 3210]
[60, 57, 58, 54, 53, 51, 39, 43, 45, 46, 30, 29, 27, 23, 15]
[111010, 111100, 111001, 110011, 110101, 110110, 101110, 101101,
↳ 101011, 100111, 001111, 010111, 011011, 011101, 011110]
[5431, 5432, 5430, 5410, 5420, 5421, 5321, 5320, 5310, 5210,
↳ 3210, 4210, 4310, 4320, 4321]
[58, 60, 57, 51, 53, 54, 46, 45, 43, 39, 15, 23, 27, 29, 30]
[111001, 111100, 111010, 110110, 110101, 110011, 100111, 101011,
↳ 101101, 101110, 011110, 011101, 011011, 010111, 001111]
[5430, 5432, 5431, 5421, 5420, 5410, 5210, 5310, 5320, 5321,
↳ 4321, 4320, 4310, 4210, 3210]
[57, 60, 58, 54, 53, 51, 39, 43, 45, 46, 30, 29, 27, 23, 15]
[110101, 110110, 111010, 111100, 111001, 110011, 100111, 101011,
↳ 101101, 101110, 011110, 011101, 011011, 010111, 001111]
[5420, 5421, 5431, 5432, 5430, 5410, 5210, 5310, 5320, 5321,
↳ 4321, 4320, 4310, 4210, 3210]
[53, 54, 58, 60, 57, 51, 39, 43, 45, 46, 30, 29, 27, 23, 15]
```



#### Παράδειγμα 4

Αν ο χρήστης του προγράμματος δώσει:

```
python transposition_graphs.py 2 6 bts
```

το πρόγραμμά σας θα πρέπει να εμφανίσει ακριβώς τα παρακάτω:

```
['000000-+', '00000-0-', '00000--0', '0000-+00', '0000-0+0',
↳ '0000-00+', '000-000-', '000-00-0', '000-0-00', '000--000',
↳ '00-+0000', '00-0+000', '00-00+00', '00-000+0', '00-0000+',
↳ '0-00000-', '0-0000-0', '0-000-00', '0-00-000', '0-0-0000',
↳ '0--00000', '-+000000', '-0+00000', '-00+0000', '-000+000',
↳ '-0000+00', '-00000+0', '-000000+']
[11111100, 11111010, 11111001, 11110011, 11110101, 11110110,
↳ 11101110, 11101101, 11101011, 11100111, 11001111, 11010111,
↳ 11011011, 11011101, 11011110, 10111110, 10111101, 10111011,
↳ 10110111, 10101111, 10011111, 00111111, 01011111, 01101111,
↳ 01110111, 01111011, 01111101, 01111110]
[765432, 765431, 765430, 765410, 765420, 765421, 765321, 765320,
↳ 765310, 765210, 763210, 764210, 764310, 764320, 764321,
↳ 754321, 754320, 754310, 754210, 753210, 743210, 543210,
↳ 643210, 653210, 654210, 654310, 654320, 654321]
[252, 250, 249, 243, 245, 246, 238, 237, 235, 231, 207, 215, 219,
↳ 221, 222, 190, 189, 187, 183, 175, 159, 63, 95, 111, 119,
↳ 123, 125, 126]
['0000000--', '00000-+0', '00000-0+', '0000-00-', '0000-0-0',
↳ '0000--00', '000-+000', '000-0+00', '000-00+0', '000-000+',
↳ '00-0000-', '00-000-0', '00-00-00', '00-0-000', '00--0000',
↳ '0-+00000', '0-0+0000', '0-00+000', '0-000+00', '0-0000+0',
↳ '0-00000+', '-000000-', '-00000-0', '-0000-00', '-000-000',
↳ '-00-0000', '-0-00000', '--000000']
[11111100, 11111001, 11111010, 11110110, 11110101, 11110011,
↳ 11100111, 11101011, 11101101, 11101110, 11011110, 11011101,
↳ 11011011, 11010111, 11001111, 10011111, 10101111, 10110111,
↳ 10111011, 10111101, 10111110, 01111110, 01111101, 01111011,
↳ 01110111, 01101111, 01011111, 00111111]
[765432, 765430, 765431, 765421, 765420, 765410, 765210, 765310,
↳ 765320, 765321, 764321, 764320, 764310, 764210, 763210,
↳ 743210, 753210, 754210, 754310, 754320, 754321, 654321,
↳ 654320, 654310, 654210, 653210, 643210, 543210]
[252, 249, 250, 246, 245, 243, 231, 235, 237, 238, 222, 221, 219,
↳ 215, 207, 159, 175, 183, 187, 189, 190, 126, 125, 123, 119,
↳ 111, 95, 63]
```

#### Παράδειγμα 5

Αν ο χρήστης του προγράμματος δώσει:

```
python transposition_graphs.py 2 6 dfs 250
```

το πρόγραμμά σας θα πρέπει να εμφανίσει ακριβώς τα παρακάτω:

```
[11111010, 11111100, 11111001, 11110011, 11110101, 11110110,  
  ↳ 11101110, 11101101, 11101011, 11100111, 11001111, 11010111,  
  ↳ 11011011, 11011101, 11011110, 10111110, 10111101, 10111011,  
  ↳ 10110111, 10101111, 10011111, 00111111, 01011111, 01101111,  
  ↳ 01110111, 01111011, 01111101, 01111110]  
[765431, 765432, 765430, 765410, 765420, 765421, 765321, 765320,  
  ↳ 765310, 765210, 763210, 764210, 764310, 764320, 764321,  
  ↳ 754321, 754320, 754310, 754210, 753210, 743210, 543210,  
  ↳ 643210, 653210, 654210, 654310, 654320, 654321]  
[250, 252, 249, 243, 245, 246, 238, 237, 235, 231, 207, 215, 219,  
  ↳ 221, 222, 190, 189, 187, 183, 175, 159, 63, 95, 111, 119,  
  ↳ 123, 125, 126]
```

Καλή Επιτυχία!