Αλγόριθμοι και Πολυπλοκότητα

Αρχοντία Γιαννοπούλου
Όλγα Φουρτουνέλλη

Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών

Δυναμικός Προγραμματισμός

Ομοιότητα Συμβολοσειρών

Αλγοριθμικά Μοντέλα

Απληστία. Χτίσε μια λύση σταδιακά, βελτιστοποιώντας μυωπικά κάποιο τοπικό κριτήριο.

Διαίρει και κυρίευε. Διάσπασε ένα πρόβλημα σε δύο υποπροβλήματα, λύσε κάθε υποπρόβλημα ανεξάρτητα, και συνδύασε τις λύσεις των υποπροβλημάτων για να δημιουργήσεις την λύση του αρχικού προβλήματος.

Αλγοριθμικά Μοντέλα

Απληστία. Χτίσε μια λύση σταδιακά, βελτιστοποιώντας μυωπικά κάποιο τοπικό κριτήριο.

Διαίρει και κυρίευε. Διάσπασε ένα πρόβλημα σε δύο υποπροβλήματα, λύσε κάθε υποπρόβλημα ανεξάρτητα, και συνδύασε τις λύσεις των υποπροβλημάτων για να δημιουργήσεις την λύση του αρχικού προβλήματος.

Δυναμικός προγραμματισμός.

Αλγοριθμικά Μοντέλα

Απληστία. Χτίσε μια λύση σταδιακά, βελτιστοποιώντας μυωπικά κάποιο τοπικό κριτήριο.

Διαίρει και κυρίευε. Διάσπασε ένα πρόβλημα σε δύο υποπροβλήματα, λύσε κάθε υποπρόβλημα ανεξάρτητα, και συνδύασε τις λύσεις των υποπροβλημάτων για να δημιουργήσεις την λύση του αρχικού προβλήματος.

Δυναμικός προγραμματισμός. Διάσπασε ένα πρόβλημα σε μια σειρά από επικαλυπτόμενα υποπροβλήματα, και δόμησε σωστές λύσεις για όλο και μεγαλύτερα υποπροβλήματα.

Σύνοψη

- Χαρακτηρισμός δομής του προβλήματος.
- Αναδρομικός ορισμός της τιμής της βέλτιστης λύσης.
- Υπολογισμός της τιμής της βέλτιστης λύσης.
- Κατασκευή της βέλτιστης λύσης από την υπολογισμένη πληροφορία.

Μέγιστη Κοινή Υπακολουθία

Δεδομένα: δύο ακολουθίες/συμβολοσειρές $x = x_1 x_2 \cdots x_m$ και $y = y_1 y_2 \ldots y_n$

Ζητούμενο: Το μέγιστο μήκος μίας ακολουθίας $z_1 z_2 \cdots z_k$ η οποία είναι ταυτόχρονα υπακολουθία των x και y

Αλγόριθμοι Πολυπλοκότητα

Δομή του Προβλήματος

$\mathrm{OPT}(i, j) =$ μέγιστο μήκος μίας κοινής υπακολουθίας των $x_1 x_2 \cdots x_i$ και $y_1 y_2 \cdots y_j$.

Δομή του Προβλήματος

$\mathrm{OPT}(i,j)$ = μέγιστο μήκος μίας κοινής υπακολουθίας των $x_1 x_2 \cdots x_i$ και $y_1 y_2 \cdots y_j$.

- $x_i = y_j = z_k$.

- $x_i \neq y_j$.

Δομή του Προβλήματος

$\mathrm{OPT}(i,j)$ = μέγιστο μήκος μίας κοινής υπακολουθίας των $x_1 x_2 \cdots x_i$ και $y_1 y_2 \cdots y_j$.

- $x_i = y_j = z_k$.
  Τότε το τελευταίο στοιχείο της μέγιστης κοινής υπακολουθίας είναι το $z_k = x_i = y_j$ και η υπόλοιπη μέγιστη κοινή υπακολουθία είναι υπακολουθία των $x_1 x_2 \cdots x_{i-1}$ και $y_1 y_2 \cdots y_{j-1}$.

- $x_i \neq y_j$.

Δομή του Προβλήματος

$\mathrm{OPT}(i,j)$ = μέγιστο μήκος μίας κοινής υπακολουθίας των $x_1 x_2 \cdots x_i$ και $y_1 y_2 \cdots y_j$.

- $x_i = y_j = z_k$.
  Τότε το τελευταίο στοιχείο της μέγιστης κοινής υπακολουθίας είναι το $z_k = x_i = y_j$ και η υπόλοιπη μέγιστη κοινή υπακολουθία είναι υπακολουθία των $x_1 x_2 \cdots x_{i-1}$ και $y_1 y_2 \cdots y_{j-1}$.

- $x_i \neq y_j$.
  Τότε το πολύ ένα από τα x_i και y_j μπορεί να περιέχεται στη μέγιστη κοινή υπακολουθία και άρα η μέγιστη κοινή υπακολουθία τους είναι είτε υπακολουθία των $x_1 x_2 \cdots x_i$ και $y_1 y_2 \cdots y_{j-1}$ είτε των $x_1 x_2 \cdots x_{i-1}$ και $y_1 y_2 \cdots y_j$

Δομή του Προβλήματος

$\mathrm{OPT}(i,j)$ = μέγιστο μήκος μίας κοινής υπακολουθίας των $x_1x_2\cdots x_i$ και $y_1y_2\cdots y_j$.

- $x_i = y_j = z_k$.
  Τότε το τελευταίο στοιχείο της μέγιστης κοινής υπακολουθίας είναι το $z_k = x_i = y_j$ και η υπόλοιπη μέγιστη κοινή υπακολουθία είναι υπακολουθία των $x_1x_2\cdots x_{i-1}$ και $y_1y_2\cdots y_{j-1}$.

- $x_i \neq y_j$.
  Τότε το πολύ ένα από τα x_i και y_j μπορεί να περιέχεται στη μέγιστη κοινή υπακολουθία και άρα η μέγιστη κοινή υπακολουθία τους είναι είτε υπακολουθία των $x_1x_2\cdots x_i$ και $y_1y_2\cdots y_{j-1}$ είτε των $x_1x_2\cdots x_{i-1}$ και $y_1y_2\cdots y_j$

$$\mathrm{OPT}(i,j) = \begin{cases} 0 & i = 0 \text{ ή } j = 0 \\ 1 + \mathrm{OPT}(i-1, j-1) & i,j > 1 \text{ και } x_i = y_j \\ \max\{\mathrm{OPT}(i-1,j), \mathrm{OPT}(i,j-1)\} & i,j > 1 \text{ και } x_i \neq y_j \end{cases}$$

Αλγόριθμος

ΜΚΥ$(m, n, x_1x_2 \ldots x_m, y_1y_2 \ldots y_n)$
για $j = 0$ έως n
    $M[0, j] = 0$
για $i = 0$ έως m
    $M[i, 0] = 0$

για $i = 1$ εως m
    για $j = 1$ εως n
        εάν $x_i = y_j$
            $M[i, j] = 1 + M[i - 1, j - 1]$
        αλλιώς
            $M[i, j] = \max(M[i, j - 1], M[i - 1, j])$

Αλγόριθμος

ΜΚΥ$(m, n, x_1 x_2 \ldots x_m, y_1 y_2 \ldots y_n)$
για $j = 0$ έως n
    $M[0, j] = 0$
για $i = 0$ έως m
    $M[i, 0] = 0$

για $i = 1$ εως m
    για $j = 1$ εως n
        εάν $x_i = y_j$
            $M[i, j] = 1 + M[i - 1, j - 1]$
        αλλιώς
            $M[i, j] = \max(M[i, j - 1], M[i - 1, j])$

Πολυπλοκότητα:

Αλγόριθμος

ΜΚΥ$(m, n, x_1x_2 \ldots x_m, y_1y_2 \ldots y_n)$
για $j = 0$ έως n
    $M[0, j] = 0$
για $i = 0$ έως m
    $M[i, 0] = 0$

για $i = 1$ εως m
    για $j = 1$ εως n
        εάν $x_i = y_j$
            $M[i, j] = 1 + M[i - 1, j - 1]$
        αλλιώς
            $M[i, j] = \max(M[i, j - 1], M[i - 1, j])$

Πολυπλοκότητα: $\Theta(mn)$.

Ομοιότητα συμβολοσειρών

Δοσμένων δύο συμβολοσειρών, πόσο όμοιες είναι;

Για παράδειγμα οι λέξεις

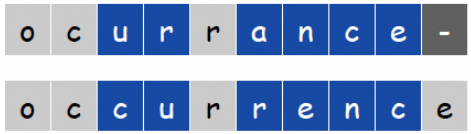
- ocurrance
- occurence

Ομοιότητα συμβολοσειρών

Δοσμένων δύο συμβολοσειρών, πόσο όμοιες είναι;

Για παράδειγμα οι λέξεις

- ocurrance
- occurence

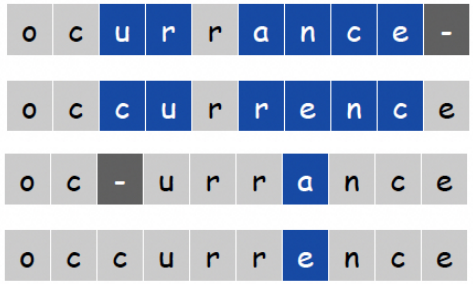


Ομοιότητα συμβολοσειρών

Δοσμένων δύο συμβολοσειρών, πόσο όμοιες είναι;

Για παράδειγμα οι λέξεις

- ocurrance
- occurence

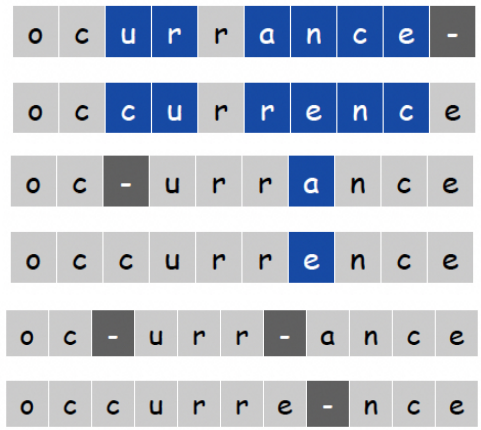


Ομοιότητα συμβολοσειρών

Δοσμένων δύο συμβολοσειρών, πόσο όμοιες είναι;

Για παράδειγμα οι λέξεις

- ocurrance
- occurence



Απόσταση επεξεργασίας

Εφαρμογές

- Αποτελεί βάση για την εντολή diff του unix
- Αναγνώριση φωνής
- Υπολογιστική Βιολογία

Απόσταση επεξεργασίας

Εφαρμογές

- Αποτελεί βάση για την εντολή diff του unix
- Αναγνώριση φωνής
- Υπολογιστική Βιολογία

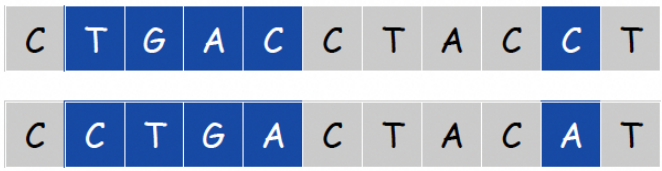
Απόσταση επεξεργασίας: [Levenshtein 1966, Needleman-Wunsch 1970]
Κόστος κενού δ, κόστος σύγκρουσης α
Κόστος $=$ άθροισμα κόστους κενού και κόστους σύγκρουσης

Απόσταση επεξεργασίας

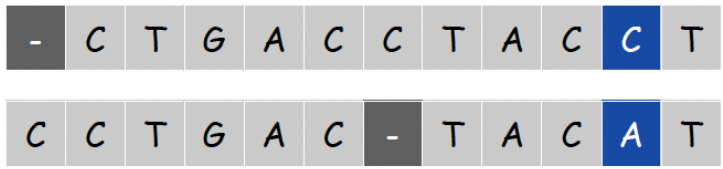
Εφαρμογές

- Αποτελεί βάση για την εντολή diff του unix
- Αναγνώριση φωνής
- Υπολογιστική Βιολογία

Απόσταση επεξεργασίας: [Levenshtein 1966, Needleman-Wunsch 1970]
Κόστος κενού δ, κόστος σύγκρουσης α
Κόστος $=$ άθροισμα κόστους κενού και κόστους σύγκρουσης



$\alpha_{TC} + \alpha_{GT} + \alpha_{AG} + 2\alpha_{CA}$



$2\delta + \alpha_{CA}$

Ευθυγράμμιση συμβολοσειρών

- Στόχος: Δοσμένων δύο συμβολοσειρών $X = x_1 x_2 \dots x_m$ και $Y = y_1 y_2 \dots y_n$ να βρεθεί ευθυγράμμιση ελάχιστου κόστους
- Μία ευθυγράμμιση M είναι ένα σύνολο από διατεταγμένα ζεύγη $x_i - y_j$ τέτοια ώστε κάθε στοιχείο βρίσκεται σε το πολύ ένα ζεύγος και τα ζεύγη δε διασταυρώνονται
- Τα ζευγάρια $x_i - y_j$ και $x_{i'} - y_{j'}$ διασταυρώνονται εάν $i < i'$ αλλά $j > j'$

$$\text{κόστος}(M) = \sum_{(x_i - y_j) \in M,\ x_i \neq y_j} \alpha_{x_i - y_j} + \sum_{i : x_i \notin M} \delta + \sum_{j : y_j \notin M} \delta$$

CTACCG **vs.** TACATG.

$M = x_2\text{-}y_1, x_3\text{-}y_2, x_4\text{-}y_3, x_5\text{-}y_4, x_6\text{-}y_6.$

x_1	x_2	x_3	x_4	x_5		x_6
C	T	A	C	C	-	G
-	T	A	C	A	T	G
y_1	y_2	y_3	y_4	y_5	y_6	

Δομή προβλήματος

$\mathrm{OPT}(i, j) =$ ελάχιστο κόστος ευθυγράμμισης των συμβολοσειρών $x_1 x_2 \ldots x_i$ και $y_1 y_2 \cdots y_j$.

Δομή προβλήματος

$\mathrm{OPT}(i, j) =$ ελάχιστο κόστος ευθυγράμμισης των συμβολοσειρών $x_1 x_2 \ldots x_i$ και $y_1 y_2 \cdots y_j$.

- Το OPT ταιριάζει τα x_i και y_j.

Δομή προβλήματος

$\mathrm{OPT}(i, j)$ = ελάχιστο κόστος ευθυγράμμισης των συμβολοσειρών $x_1 x_2 \ldots x_i$ και $y_1 y_2 \cdots y_j$.

- Το OPT ταιριάζει τα x_i και y_j.

- Το OPT αφήνει το x_i αταίριαστο.

Δομή προβλήματος

$\mathrm{OPT}(i, j)$ = ελάχιστο κόστος ευθυγράμμισης των συμβολοσειρών $x_1 x_2 \ldots x_i$ και $y_1 y_2 \cdots y_j$.

- Το OPT ταιριάζει τα x_i και y_j.
- Το OPT αφήνει το x_i αταίριαστο.
- Το OPT αφήνει το y_j αταίριαστο.

Δομή προβλήματος

$\mathrm{OPT}(i, j)$ = ελάχιστο κόστος ευθυγράμμισης των συμβολοσειρών $x_1 x_2 \ldots x_i$ και $y_1 y_2 \ldots y_j$.

- Το OPT ταιριάζει τα x_i και y_j.
  - Πληρώνουμε το κόστος $\alpha_{x_i - y_j}$ συν το ελάχιστο κόστος ευθυγράμμισης των $x_1 x_2 \ldots x_{i-1}$ και $y_1 y_2 \ldots y_{j-1}$.
- Το OPT αφήνει το x_i αταίριαστο.
- Το OPT αφήνει το y_j αταίριαστο.

Δομή προβλήματος

$\mathrm{OPT}(i, j)$ = ελάχιστο κόστος ευθυγράμμισης των συμβολοσειρών $x_1 x_2 \ldots x_i$ και $y_1 y_2 \ldots y_j$.

- Το OPT ταιριάζει τα x_i και y_j.
  - Πληρώνουμε το κόστος $\alpha_{x_i - y_j}$ συν το ελάχιστο κόστος ευθυγράμμισης των $x_1 x_2 \ldots x_{i-1}$ και $y_1 y_2 \ldots y_{j-1}$.
- Το OPT αφήνει το x_i αταίριαστο.
  - Πληρώνουμε το κόστος του κενού για το x_i συν το ελάχιστο κόστος ευθυγράμμισης των $x_1 x_2 \ldots x_{i-1}$ και $y_1 y_2 \ldots y_j$.
- Το OPT αφήνει το y_j αταίριαστο.

Δομή προβλήματος

$\mathrm{OPT}(i, j)$ = ελάχιστο κόστος ευθυγράμμισης των συμβολοσειρών $x_1 x_2 \ldots x_i$ και $y_1 y_2 \ldots y_j$.

- Το OPT ταιριάζει τα x_i και y_j.
  - Πληρώνουμε το κόστος $\alpha_{x_i - y_j}$ συν το ελάχιστο κόστος ευθυγράμμισης των $x_1 x_2 \ldots x_{i-1}$ και $y_1 y_2 \ldots y_{j-1}$.
- Το OPT αφήνει το x_i αταίριαστο.
  - Πληρώνουμε το κόστος του κενού για το x_i συν το ελάχιστο κόστος ευθυγράμμισης των $x_1 x_2 \ldots x_{i-1}$ και $y_1 y_2 \ldots y_j$.
- Το OPT αφήνει το y_j αταίριαστο.
  - Πληρώνουμε το κόστος του κενού για το y_j συν το ελάχιστο κόστος ευθυγράμμισης των $x_1 x_2 \ldots x_i$ και $y_1 y_2 \ldots y_{j-1}$.

Αναδρομικός τύπος

$$\mathrm{OPT}(i,j) = \begin{cases} j \cdot \delta & i = 0 \\ \min \begin{cases} \alpha_{x_i y_i} + \mathrm{OPT}(i-1, j-1), \\ \delta + \mathrm{OPT}(i-1, j), \\ \delta + \mathrm{OPT}(i, j-1) \end{cases} & \text{αλλιώς} \\ i \cdot \delta & j = 0 \end{cases}$$

Αλγόριθμος

Ενθυγράμμιση$(m, n, x_1x_2 \ldots x_m, y_1y_2 \ldots y_n, \delta, \alpha)$
για $i = 0$ έως m
    $M[0, i] = i\delta$
για $j = 0$ έως n
    $M[j, 0] = j\delta$

για $i = 1$ εως m
    για $j = 1$ εως n
        $M[i, j] = \min(\alpha_{x_i - y_j} + M[i-1, j-1],$
        $\delta + M[i-1, j],$
        $\delta + M[i, j-1])$

επίστρεψε $M[m, n]$

Πολυπλοκότητα: $\Theta(mn)$ χρόνο και χώρο.
Οι περισσότερες λέξεις έχουν λίγα γράμματα.
Στην Υπολογιστική Βιολογία $m = n = 100,000$.
Ο χώρος που απαιτείται για την αποθήκευση του πίνακα είναι πολύ μεγάλος.

Παράδειγμα

Για ευκολία $\delta = 1$ και $a_{xy} = 1$ για $x \neq y$

	ϵ	G	C	T	A	T	G	C	C	A	C	G	C
ϵ	0	1	2	3	4	5	6	7	8	9	10	11	12
G	1												
C	2												
G	3												
T	4												
A	5												
T	6												
G	7												
C	8												
A	9												
C	10												
G	11												
C	12												

Παράδειγμα

Για ευκολία $\delta = 1$ και $a_{xy} = 1$ για $x \neq y$

	ϵ	G	C	T	A	T	G	C	C	A	C	G	C
ϵ	0	1	2	3	4	5	6	7	8	9	10	11	12
G	1	0	1	2	3	4	5	6	7	8	9	10	11
C	2	1	0	1	2	3	4	5	6	7	8	9	10
G	3	2	1	1	2	3	3	4	5	6	7	8	9
T	4	3	2	1	2	2	3	4	5	6	7	8	9
A	5	4	3	2	1	2	3	4	5	5	6	7	8
T	6	5	4	3	2	1	2	3	4	5	6	7	8
G	7	6	5	4	3	2	1	2	3	4	5	6	7
C	8	7	6	5	4	3	2	1	2	3	4	5	6
A	9	8	7	6	5	4	3	2	2	2	3	4	5
C	10	9	8	7	6	5	4	3	2	3	2	3	4
G	11	10	9	8	7	6	5	4	3	3	3	2	3
C	12	11	10	9	8	7	6	5	4	4	3	3	2

Ευθυγράμμιση συμβολοσειρών σε γραμμικό χώρο

Μπορούμε να αποφύγουμε τον τετραγωνικό χώρο;

Υπολογίζουμε τη βέλτιστη **τιμή** σε $\mathcal{O}(m+n)$ χώρο και $\mathcal{O}(mn)$ χρόνο.

- Υπολογισμός $\mathrm{OPT}(i, \star)$ από το $\mathrm{OPT}(i-1, \star)$
- Δεν είναι εύκολο να ανακτήσουμε την ευθυγράμμιση

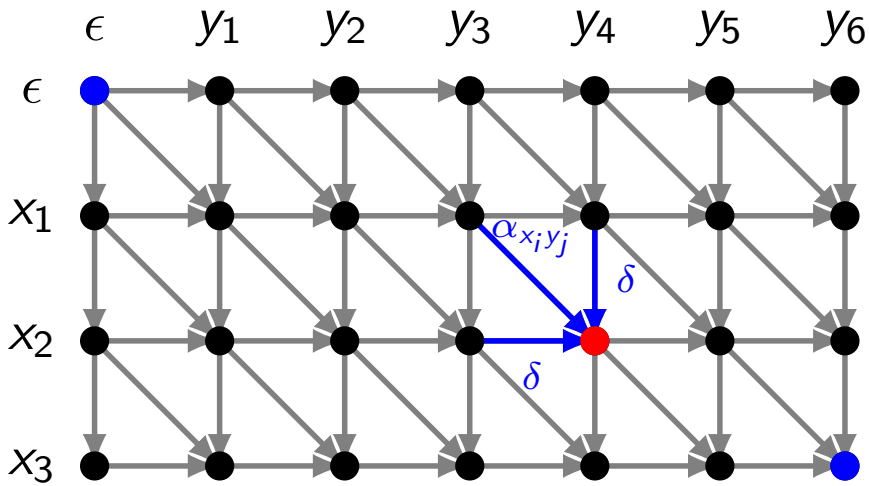
[Hirschberg 1975] Βέλτιστη ευθυγράμμιση σε $\mathcal{O}(m+n)$ χώρο και $\mathcal{O}(mn)$ χρόνο.

- Συνδυασμός τεχνικών Διαίρει-και-Κυρίευε και Δυναμικού Προγραμματισμού
- Εμπνευσμένο από την Υπολογιστική Πολυπλοκότητα

Ευθυγράμμιση συμβολοσειρών σε γραμμικό χώρο

Γράφημα Απόστασης Επεξεργασίας

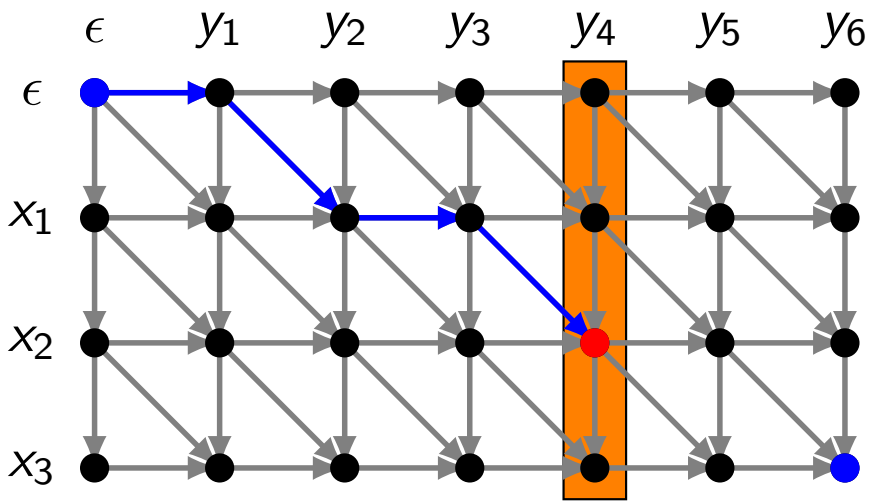
- Έστω $f(i,j)$ το συντομότερο μονοπάτι από τη κορυφή $(0,0)$ έως την (i,j)
- $f(i,j) = \mathrm{OPT}(i,j)$



Ευθυγράμμιση συμβολοσειρών σε γραμμικό χώρο

Γράφημα Απόστασης Επεξεργασίας

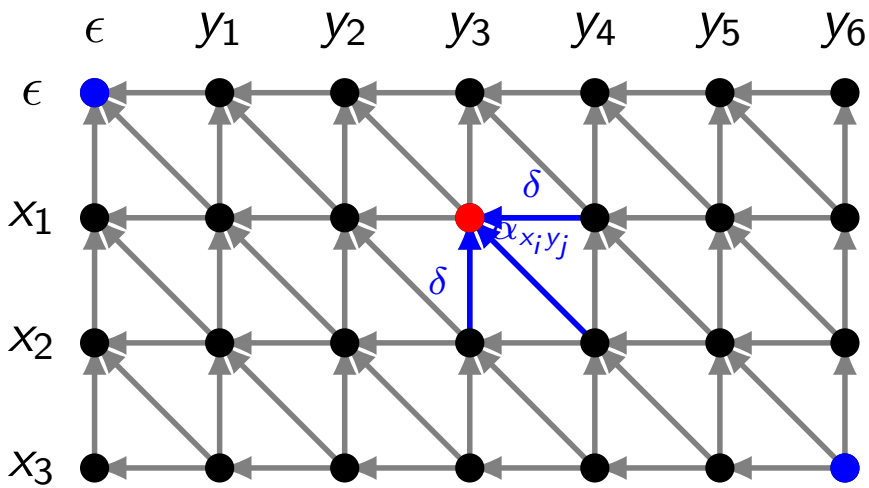
- Έστω $f(i,j)$ το συντομότερο μονοπάτι από τη κορυφή $(0,0)$ έως την (i,j)
- $f(i,j) = \mathrm{OPT}(i,j)$
- Μπορούμε να υπολογίσουμε το $f(\cdot,j)$ σε $\mathcal{O}(m+n)$ χώρο και $\mathcal{O}(mn)$ χρόνο για οποιοδήποτε j



Ευθυγράμμιση συμβολοσειρών σε γραμμικό χώρο

Γράφημα Απόστασης Επεξεργασίας

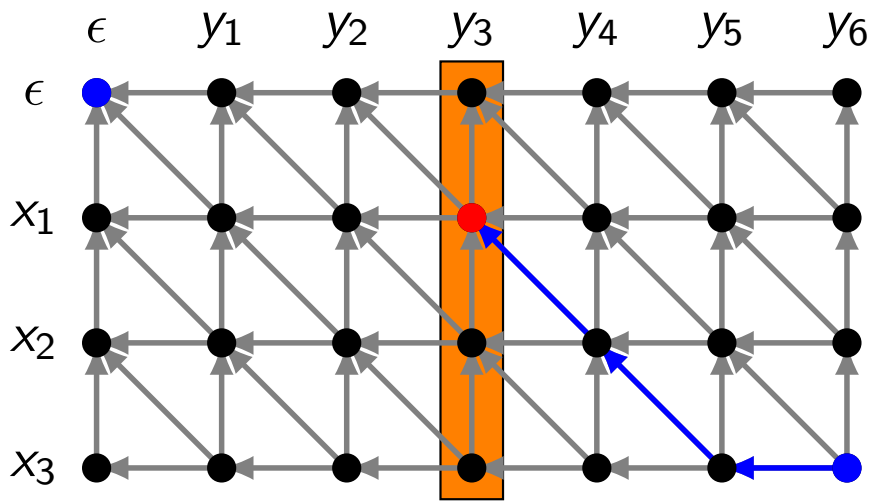
- Έστω $g(i,j)$ το συντομότερο μονοπάτι από τη κορυφή (i,j) έως την (m,n)



Ευθυγράμμιση συμβολοσειρών σε γραμμικό χώρο

Γράφημα Απόστασης Επεξεργασίας

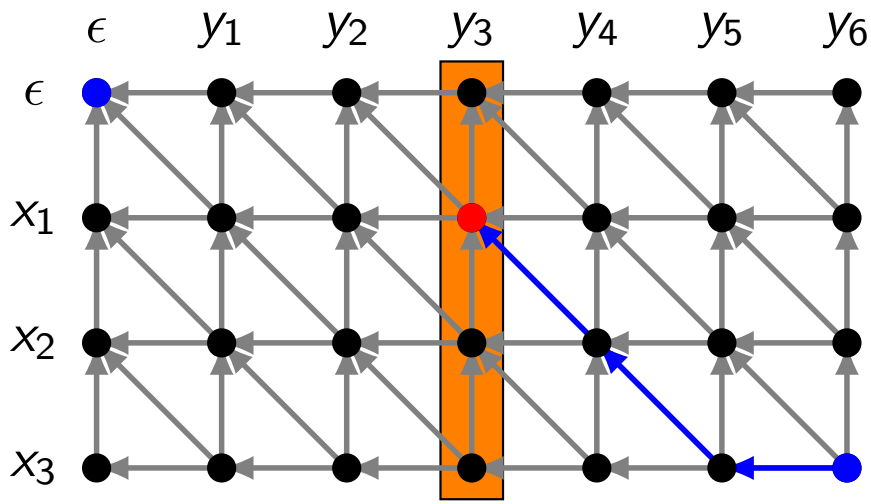
- Έστω $g(i, j)$ το συντομότερο μονοπάτι από τη κορυφή (i, j) έως την (m, n)
- Μπορεί να υπολογιστεί αντιστρέφοντας την κατεύθυνση των ακμών και ανταλλάσσοντας την $(0, 0)$ με την (m, n)



Ευθυγράμμιση συμβολοσειρών σε γραμμικό χώρο

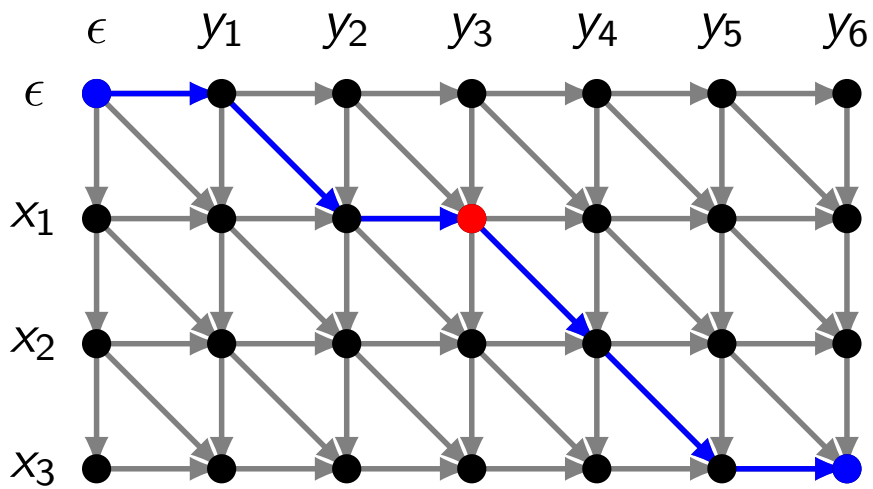
Γράφημα Απόστασης Επεξεργασίας

- Έστω $g(i, j)$ το συντομότερο μονοπάτι από τη κορυφή (i, j) έως την (m, n)
- Μπορεί να υπολογιστεί αντιστρέφοντας την κατεύθυνση των ακμών και ανταλλάσσοντας την $(0, 0)$ με την (m, n)
- Πολυπλοκότητα όπως πριν



Ευθυγράμμιση συμβολοσειρών σε γραμμικό χώρο

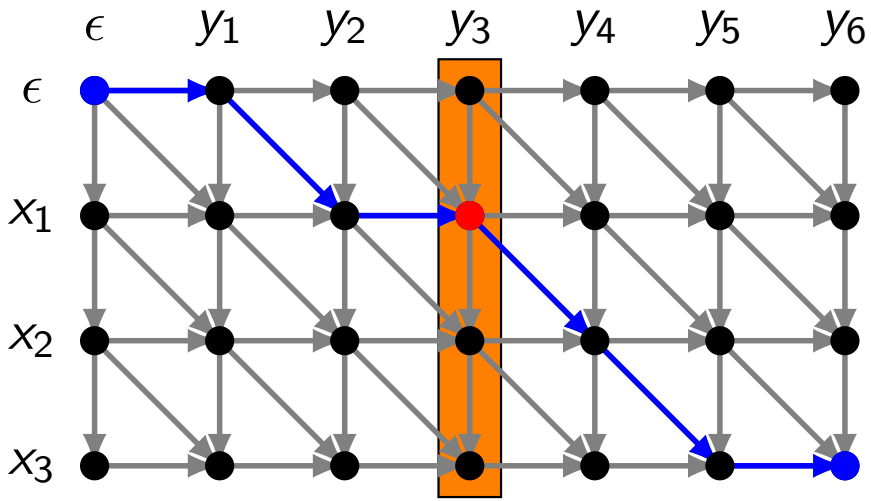
Το κόστος του συντομότερου μονοπατιού που περιέχει την (i, j) είναι $f(i, j) + g(i, j)$



Ευθυγράμμιση συμβολοσειρών σε γραμμικό χώρο

Το κόστος του συντομότερου μονοπατιού που περιέχει την (i, j) είναι $f(i, j) + g(i, j)$

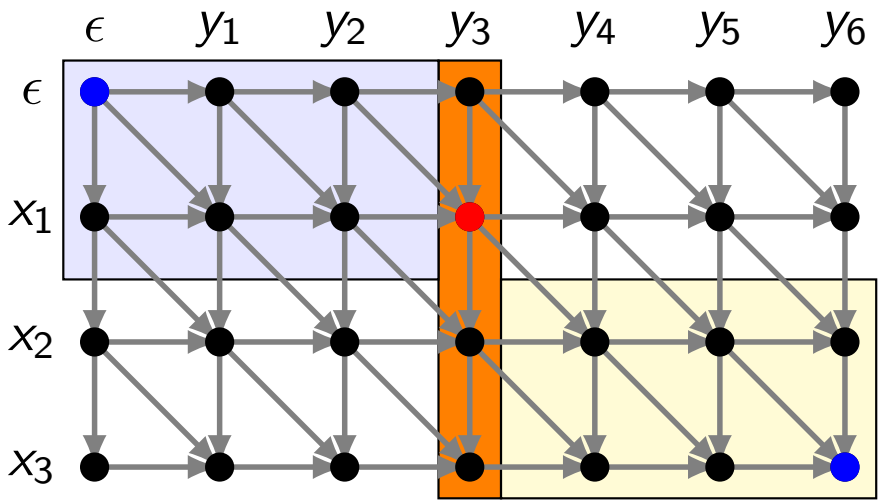
Αν q είναι ο δείκτης για τον οποίο η ποσότητα $f(q, \frac{n}{2}) + g(q, \frac{n}{2})$ ελαχιστοποιείται τότε το συντομότερο μονοπάτι από το $(0, 0)$ στο (m, n) περιέχει την $(q, \frac{n}{2})$



Ευθυγράμμιση συμβολοσειρών σε γραμμικό χώρο

Διαίρει: Βρίσκουμε q που ελαχιστοποιεί την ποσότητα $f(q, \frac{n}{2}) + g(q, \frac{n}{2})$ με Δυναμικό Προγραμματισμό. Ευθυγραμμίζουμε τα x_q και $y_{\frac{n}{2}}$.

Κυρίευε: Υπολογίζουμε αναδρομικά τη βέλτιστη ευθυγράμμιση σε κάθε ένα από τα δύο υποπροβλήματα.



Ευθυγράμμιση συμβολοσειρών: Χρονική Πολυπλοκότητα

Έστω $T(m, n)$ ο μέγιστος χρόνος που χρειάζεται ο αλγόριθμος με είσοδο δύο συμβολοσειρές μήκους m και n αντίστοιχα. Τότε $T(m, n) = \mathcal{O}(mn)$.

Με επαγωγή στο n.

Άσκηση!