



ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

Τμήμα Πληροφορικής

ΕΠΛ 232 – Προγραμματιστικές Τεχνικές και Εργαλεία

ΑΣΚΗΣΗ 2 – Συμβολοσειρές, Συναρτήσεις, Αρχεία και Δείκτες

Διδάσκων: Δημήτρης Ζεϊναλιπούρ

Υπεύθυνοι Εργαστηρίων: Πύρρος Μπράτσкас & Παύλος Αντωνίου

Ημερομηνία Ανάθεσης: Τρίτη, 25 Σεπτεμβρίου 2018

Ημερομηνία Παράδοσης: Τρίτη, 9 Οκτωβρίου 2018 ώρα 09:00 (14 ημέρες)

(ο κώδικας να υποβληθεί σε zip μέσω του Moodle)

<https://www.cs.ucy.ac.cy/courses/EPL232>

I. Στόχος

Στην άσκηση αυτή θα ασχοληθούμε σε περισσότερο βάθος με τη γλώσσα προγραμματισμού C. Η άσκηση απαιτεί μεταξύ άλλων χρήση συμβολοσειρών, συναρτήσεων, αρχείων και δεικτών. Αποτελείται από τρία (3) θέματα. Υλοποιήστε το κάθε θέμα σε ξεχωριστό αρχείο C.

II. Θέμα 1: Σύγκριση Αρχείων με την `fcompare`

Φτιάξτε πρόγραμμα, με όνομα `fcompare.c`, που θα διαβάζει δυο ASCII αρχεία, θα τα συγκρίνει και θα εμφανίζει στην οθόνη τις γραμμές που διαφέρουν. Το πρόγραμμα πρέπει να δουλεύει ορθά για αρχεία που περιέχουν οποιουδήποτε ASCII χαρακτήρες, και να υποστηρίζει παραμέτρους στη γραμμή εντολών ως ακολούθως:

Σύνταξη: `fcompare [options] file1 file2`

Options:

`/nc` : Αγνόησε διάκριση κεφαλαίων-πεζών στη σύγκριση

`/s` : Επίστρεψε μόνο την πρώτη και την τελευταία γραμμή από ένα σύνολο συνεχόμενων διαφορετικών γραμμών

`/ln` : Τύπωσε και τους αριθμούς των διαφορετικών γραμμών στην έξοδο

Σημειώσεις:

A) Οι επιλογές (options) μπορούν να δηλωθούν με οποιαδήποτε σειρά

B) Αν τα αρχεία έχουν διαφορετικό αριθμό γραμμών, η σύγκριση σταματά όταν τελειώσει το μικρότερο αρχείο. Στο αποτέλεσμα εμφανίζεται και ο αριθμός των υπολειπόμενων γραμμών του μεγαλύτερου αρχείου δηλαδή ένα μήνυμα της μορφής:

```
"* The second file has 2 more line(s) than the first."
```

Γ) Τα options τα οποία δεν υποστηρίζονται (π.χ., `/g`) να αγνοούνται.

Παραδείγματα Κατανόησης:

Έστω ότι οι γραμμές δύο αρχείων `file1` και `file2` συγκρίνονται ως εξής:

1: ίδιες

2: διαφορετικές

3: διαφορετικές

4: διαφορετικές

5: ίδιες

Τότε η εντολή `fcompare /s /ln file1 file2` θα επιστρέψει

2: «περιεχόμενο γραμμής του αρχείου 1»

2: «περιεχόμενο γραμμής του αρχείου 2»

Down to

4: «περιεχόμενο γραμμής του αρχείου 1»

4: «περιεχόμενο γραμμής του αρχείου 2»

Έστω ότι οι γραμμές των αρχείων συγκρίνονται ως εξής:

1: ίδιες

2: διαφορετικές

3: ίδιες

4: διαφορετικές

5: ίδιες

Τότε η εντολή `fcompare /ln /s file1 file2`
θα επιστρέψει

2: «περιεχόμενο γραμμής του αρχείου 1»

2: «περιεχόμενο γραμμής του αρχείου 2»

4: «περιεχόμενο γραμμής του αρχείου 1»

4: «περιεχόμενο γραμμής του αρχείου 2»

Υποδειγματικές Εκτελέσεις

Για να δοκιμάσετε το πρόγραμμα σας μπορείτε να ανακτήσετε από την ιστοσελίδα το αρχείο `as2-supplementary.zip`, κάτω από το σύνδεσμο της Εργασίας 2, το οποίο περιέχει στον κατάλογο `2.1-supplementary` διάφορα δοκιμαστικά αρχεία εισόδου. Μερικά παραδείγματα εκτέλεσης με τα αρχεία αυτά ακολουθούν:

```
$./as2-1 /ln /s test11.txt test12.txt
```

```
3: f1 l3
3: f2 l3
Down to
6: f1 l6
6: f2 l6
8: f1 l8
8: f2 l8
Down to
9: f1 l9
9: f2 l9
11: aSd l11
11: ASD l11
Down to
12: f1 l12
12: f2 l12
15: f1 l15
15: f2 l15
Down to
17: f1 l17
17: f2 l17
```

```
$./as2-1 /ln /s test21.txt test22.txt
```

```
1: f1 l1
1: f2 l1
6: f1 l6
6: f2 l6
11: f1 l11
11: f2 l11
14: f1 l14
14: f2 l14
17: ASd
17: aSd
20: f1 l20
20: f2 l20
```

	test11.txt	test12.txt
1 :	f l	f l
2 :	f l2	f l2
3 :	f1 l3	f2 l3
4 :	f1 l4	f2 l4
5 :	f1 l5	f2 l5
6 :	f1 l6	f2 l6
7 :	f l7	f l7
8 :	f1 l8	f2 l8
9 :	f1 l9	f2 l9
10:	f l10	f l10
11:	aSd l11	ASD l11
12:	f1 l12	f2 l12
13:	f l13	f l13
14:	f l14	f l14
15:	f1 l15	f2 l15
16:	f1 l16	f2 l16
17:	f1 l17	f2 l17

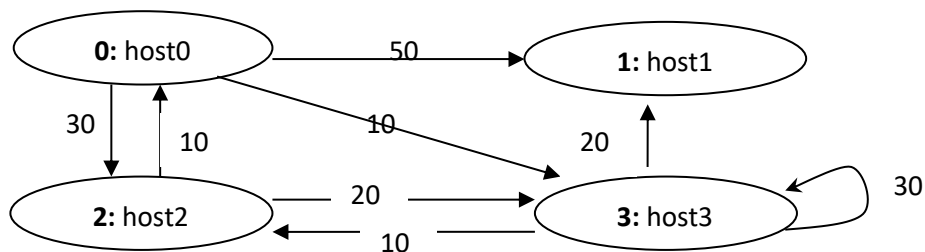
	Test21.txt	Test22.txt
1 :	f1 l1	f2 l1
2 :	f l	f l
3 :	a	a
4 :	a	a
5 :	a	a
6 :	f1 l6	f2 l6
7 :	f l	f l
8 :	f l	f l
9 :	a	a
10:	a	a
11:	f1 l11	f2 l11
12:	f l	f l
13:	f l	f l
14:	f1 l14	f2 l14
15:	f l	f l
16:	f l	f l
17:	ASd	aSd
18:	a	a
19:	s	s
20:	f1 l20	f2 l20
21:	f l	f l
22:	f l	f l

III. Θέμα 2: Υλοποίηση Αλγόριθμου Pagerank

Στο θέμα αυτό θα υλοποιήσετε ένα αλγόριθμο που υπολογίζει τη σημασία ενός φιλοξενητή ιστοσελίδων (web host) με βάση τη σημασία άλλων φιλοξενητών που οι ιστοσελίδες τους έχουν υπερσυνδέσμους (hyperlinks) προς αυτόν. Ο αλγόριθμος αυτός είναι μια απλοποιημένη έκδοση του αλγορίθμου PageRank που χρησιμοποιεί για την ταξινόμηση των ιστοσελίδων η μηχανή αναζήτησης Google.

Για να υπολογίσουμε τη σημασία κάθε φιλοξενητή ιστοσελίδων, μπορούμε να θεωρήσουμε την εξής λογική δομή δεδομένων για τον παγκόσμιο Ιστό (WWW): Ένα διδιάστατο πίνακα διαστάσεων $m \times m$ που καταγράφει στο στοιχείο (i, j) αν η ιστοσελίδα (web-page) i έχει υπερσύνδεσμο (web-link) προς τη σελίδα j , όπου m είναι ο συνολικός αριθμός σελίδων. Για να απλοποιήσουμε την περιγραφή θεωρήστε μια απλουστευμένη έκδοση του εν λόγω $m \times m$ πίνακα η οποία αποθηκεύει μόνο τις συσχετίσεις μεταξύ των φιλοξενητών-διαθετών (Web Servers). Συγκεκριμένα, θεωρείστε διδιάστατο πίνακα $n \times n$, όπου n είναι ο συνολικός αριθμός φιλοξενητών (web-hosts) στον Ιστό, που καταγράφει στο στοιχείο (i, j) τον αριθμό των υπερσυνδέσμων που ο φιλοξενητής σελίδων (web host) i έχει προς τις σελίδες που φιλοξενούνται στον j .

Για παράδειγμα, το ακόλουθο διάγραμμα παρουσιάζει τον αριθμό των συνδέσμων που έχει κάθε ένας από τους τέσσερις ($n=4$) φιλοξενητές στους υπόλοιπους φιλοξενητές.



Το διάγραμμα αυτό μπορεί να αναπαρασταθεί με το ακόλουθο 3×3 πίνακα:

	0	1	2	3
0	0	50	30	10
1	0	0	0	0
2	10	0	0	20
3	0	20	10	30

Πιο συγκεκριμένα, για ένα εξυπηρετητή h_i το PageRank $PR(h_i)$ υπολογίζεται ως:

$$PR(h_i) = \frac{1-d}{n} + d \sum_{h_j \in I(h_i)} \frac{PR(h_j) \cdot O_i(h_j)}{O(h_j)}$$

Όπου

- $d=0,85$ είναι ένας παράγων απόσβεσης
- n είναι ο συνολικός αριθμός φιλοξενητών (μπορεί να είναι σταθερά ή να διαβάζεται από το αρχείο)
- $I(h_i)$ είναι το σύνολο των φιλοξενητών με συνδέσμους (links) προς το φιλοξενητή h_i
- $O_i(h_j)$ είναι ο αριθμός υπερσυνδέσμων από το φιλοξενητή h_j στο φιλοξενητή h_i
- $O(h_j)$ είναι ο συνολικός αριθμός υπερσυνδέσμων από το φιλοξενητή h_j

Η γενική ιδέα του αλγορίθμου είναι ότι η «σημασία» ενός φιλοξενητή (μιας σελίδας) εξαρτάται από το πόσοι άλλοι φιλοξενητές (σελίδες) έχουν υπερσυνδέσμους προς αυτόν. Η «σημασία» που μεταφέρεται από τον j στον i φιλοξενητή είναι το κλάσμα της σημασίας του j που προκύπτει από τον ποσοστό των υπερσυνδέσμων από το j προς i . Κατ' αυτό τον τρόπο, ένας επισκέπτης του Ιστού που ακολουθεί

υπερσυνδέσμους κατά τη φυλλομέτρηση του Ιστού έχει τόση πιθανότητα να καταλήξει σε μια σελίδα όση η σημασία PR της ιστοσελίδας. Το d αντιστοιχεί στην πιθανότητα να επισκεφτεί ο επισκέπτης τη σελίδα με τυχαία μετάβαση (δηλαδή χωρίς να ακολουθήσει υπερσύνδεσμο). Για λεπτομερέστερη ανάλυση του σκεπτικού πίσω από τον υπολογισμό του PageRank, δείτε στη wikipedia.

Για την εργασία αυτή, η πληροφορία για τα ονόματα των φιλοξενιών βρίσκεται σε αρχείο `hostnames.txt` με δομή "`id name`", όπου το `id` είναι ένας αύξων αριθμός ταυτότητας του φιλοξενητή.

Οι υπερσύνδεσμοι καταγράφονται σε αρχείο `hostgraph.txt` με δομή "`src -> dest1:nlinks1 dest2:nlinks2, ..., destk:nlinksk`", όπου `src` είναι το `id` του φιλοξενητή-πηγή, `dest` είναι το `id` του προορισμού και `nlinks` είναι ο αριθμός υπερσυνδέσμων μεταξύ των δυο φιλοξενητών.

Το πρόγραμμά σας πρέπει:

A) Να υπολογίζει τις δομές $O(h_j)$, $O_i(h_j)$ και $I(h_i)$ και να τις εκτυπώνει στην οθόνη για όλες τις τιμές των i και j . Για τις δομές αυτές χρησιμοποιήστε πίνακες.

B) Να χρησιμοποιεί τα αποτελέσματα τα οποία δημιουργήσατε στο μέρος A για να υπολογίσει (επαναληπτικά) το $PR(h_i)$.

Σημείωση: Αρχικοποιήστε την τιμή του $PR(h_i)$ σε 1 για κάθε h_i . Η επανάληψη για τον υπολογισμό του διανύσματος PR πρέπει να τερματίζει αν $\|PR_i - PR_{i+1}\|_2 < 10^{-2}$, όπου ο δείκτης i αναφέρεται στην i -οστή επανάληψη και η σημειογραφία $\| \cdot \|_2$ αναφέρεται στην Ευκλείδεια απόσταση. Συγκεκριμένα, η Ευκλείδεια απόσταση μεταξύ δυο διανυσμάτων (μονοδιάστατων πινάκων) $PR_i = (p_1, p_2, \dots, p_n)$ και $PR_{i+1} = (q_1, q_2, \dots, q_n)$ δίνεται από το την τετραγωνική ρίζα του αθροίσματος των τετραγώνων των αποκλίσεων στα στοιχεία των δυο διανυσμάτων, δηλαδή :

$$\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}.$$

Εκτυπώστε τα αποτελέσματα στην οθόνη με δομή "`id name pagerank norm-pagerank`", ταξινομημένο κατά `id`. Norm-pagerank είναι το κανονικοποιημένο pagerank $\frac{PR(h_i)}{\sum_i PR(h_i)}$ με ακρίβεια δυο δεκαδικών ψηφίων.

Για να δοκιμάσετε το πρόγραμμα σας μπορείτε να ανακτήσετε από την ιστοσελίδα του μαθήματος το αρχείο **as2-supplementary.zip**, το οποίο περιέχει στον κατάλογο 2.2-supplementary, το οποίο περιέχει διάφορα δοκιμαστικά αρχεία εισόδου.

Τα ονόματα των αρχείων που περιέχουν τους κόμβους (`hostnames`) και τον γράφο (`hostgraph`) ορίζονται στη γραμμή εντολών και όλα τα αποτελέσματα εκτυπώνονται στην οθόνη.

Υποδειγματική Εκτέλεση

```
$ ./pagerank hostnames.txt hostgraph.txt
+++ Start of printing structure Oij +++
The structure of each line is:
src -> dest1:nlinks1 dest2:nlinks2 ... destk:nlinksk

0 -> 1:50 2:30 3:10
1 ->
2 -> 0:10 3:20
3 -> 1:20 2:10 3:30

--- End of printing structure Oij ---
+++ Start of printing structure Oj +++
The structure of each line is:
src ==> links_total

0 ==> 90
1 ==> 0
```

```
hostnames.txt
0 host0
1 host1
2 host2
3 host3
```

```
hostgraph.txt
0 -> 1:50 2:30 3:10
1 ->
2 -> 0:10 3:20
3 -> 1:20 2:10 3:30
```

```

2 ==> 30
3 ==> 60

--- End of printing structure Oj ---

+++ Start of printing structure Iij +++
The structure of each line is:
dest <== hosts_total hostid1 hostid2 ... hostidk

0 <== 1 2
1 <== 2 0 3
2 <== 2 0 3
3 <== 3 0 2 3

--- End of printing structure Iij ---

+++ Start of printing PR +++

host_id      host_name      host_rank      norm_host_rank
0            host0         0.062          0.14
1            host1         0.119          0.28
2            host2         0.081          0.19
3            host3         0.170          0.39

--- End of printing PR ---

```

Σημειώστε ότι τα πιο πάνω αποτελέσματα δημιουργήθηκαν από τις πιο κάτω επαναλήψεις:

```

Iteration: 0
h0 1.000
h1 1.000
h2 1.000
h3 1.000

```

```

Iteration: 1
h0 0.321
h1 0.793
h2 0.463
h3 1.124

```

```

Iteration: 2
h0 0.169
h1 0.507
h2 0.288
h3 0.807

```

```

Iteration: 3
h0 0.119
h1 0.346
h2 0.200
h3 0.560

```

```

Iteration: 4
h0 0.094
h1 0.252
h2 0.150
h3 0.400

```

```

Iteration: 5
h0 0.080
h1 0.195
h2 0.121
h3 0.302

```

```

Iteration: 6
h0 0.072
h1 0.161
h2 0.103
h3 0.242

```

```

Iteration: 7
h0 0.067
h1 0.140
h2 0.092
h3 0.205

```

hostnames.txt

```

0 host0
1 host1
2 host2
3 host3

```

hostgraph.txt

```

0 -> 1:50 2:30 3:10
1 ->
2 -> 0:10 3:20
3 -> 1:20 2:10 3:30

```

```
Iteration: 8
h0 0.064
h1 0.127
h2 0.085
h3 0.183
```

```
Iteration: 9
h0 0.062
h1 0.119
h2 0.081
h3 0.170
```

IV. Θέμα 3: Εύρεση Απόστασης Μεταξύ δυο Συμβολοσειρών

Όταν συγκρίνουμε συμβολοσειρές, τότε ο απλούστερος τρόπος απαιτεί τη σύγκριση των συμβολοσειρών στους αντίστοιχους χαρακτήρες. Για παράδειγμα, οι λέξεις «καλάμι» και «καλάθι» έχουν **μεγάλη ομοιότητα** καθώς με μία και μόνο αλλαγή γράμματος, από μ σε θ, η πρώτη λέξη μπορεί να μετατραπεί στη δεύτερη (αλλιώς λέμε ότι έχουν **μικρή απόσταση** μεταξύ τους). Τέτοιου είδους αποστάσεις μεταξύ συμβολοσειρών αναφέρονται στη βιβλιογραφία ως Lp-norm αποστάσεις (π.χ., η Ευκλείδεια απόσταση είναι η L_2)

Έστω η απόσταση SED (String Evresis Distance), η οποία ορίζεται μεταξύ οπουδήποτε δύο συμβολοσειρών τυχαίου μήκους (χωρίς να απαιτείται δηλαδή οι συμβολοσειρές αυτές να έχουν το ίδιο μήκος). Η απόσταση SED μετρά τις διαφορές μεταξύ των δύο συμβολοσειρών, όχι μόνο εκεί που υπάρχουν διαφορετικοί χαρακτήρες αλλά ακόμη και εκεί που η μια συμβολοσειρά έχει κάποιο χαρακτήρα ενώ η άλλη όχι. Για παράδειγμα η λέξη «άβουλος» μπορεί να μετατραπεί στη λέξη «άβολος» με τη διαγραφή ενός γράμματος, του «υ», ενώ ανάλογα η αντίστροφη μετατροπή γίνεται με την προσθήκη του γράμματος «υ».

Πιο αναλυτικά, η απόσταση SED μεταξύ δύο συμβολοσειρών, x και y , ορίζεται ως το μικρότερο πλήθος των αλλαγών που πρέπει να λάβουν χώρα, προκειμένου η συμβολοσειρά x να μετατραπεί στην συμβολοσειρά y . Οι αλλαγές που ενδέχεται να γίνουν είναι:

1. προσθήκη ενός γράμματος (*insertion*)
2. διαγραφή ενός γράμματος (*deletion*)
3. αλλαγή ενός γράμματος σε κάποιο άλλο (*substitution*)

Κάθε μία από τις αλλαγές αυτές μπορεί να έχει διαφορετικό βάρος στον υπολογισμό της απόστασης, ωστόσο στην απλούστερη περίπτωση οι παραπάνω τρεις περιπτώσεις είναι ισότιμες και αυξάνουν κατά μία μονάδα την απόσταση ενός ζεύγους συμβολοσειρών. Σε αυτήν την εργασία θα ασχοληθούμε με την απλή περίπτωση.

Δεδομένων 2 συμβολοσειρών x μήκους n και y μήκους m , ορίζουμε ως $D(i, j)$ την απόσταση SED μεταξύ του μήκους i προθέματος του x (δηλαδή της υποσυμβολοσειράς αποτελούμενη από τους πρώτους i χαρακτήρες της x), και του μήκους j προθέματος του y . Τότε, το $D(n, m)$ είναι η απόσταση SED μεταξύ x και y . Ο αλγόριθμος SED που βρίσκει την απόσταση μεταξύ x και y ορίζεται ως ακολούθως:

$$D(i, 0) = i$$

$$D(0, j) = j$$

Εξίσωση 1

$$D(i, j) = \min \begin{cases} D(i-1, j) + 1 \\ D(i, j-1) + 1 \\ D(i-1, j-1) + \delta(x[i-1], y[j-1]) \end{cases} \quad \delta(x[i-1], y[j-1]) = \begin{cases} 0 & \text{εάν } x[i-1] = y[j-1] \\ 1 & \text{εάν } x[i-1] \neq y[j-1] \end{cases}$$

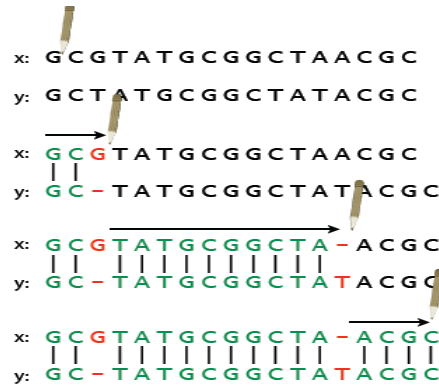
Ο υπολογισμός της απόστασης $D(n, m)$ μπορεί να γίνει αναδρομικά εφαρμόζοντας τον τύπο της Εξίσωσης 1.

Ας δούμε 3 παραδείγματα υπολογισμού της απόστασης SED.

Παράδειγμα 1:

Έστω $x = \text{"GCGTATGCGGCTAACGC"}$ και $y = \text{"GCTATGCGGCTATACGC"}$. Τότε, για να μετατραπεί το x σε y πρέπει να περάσουμε από τα στάδια που φαίνονται στην εικόνα 1:

- πρώτα πρέπει να γίνει μια διαγραφή του χαρακτήρα **G** στη 3^η θέση του **x**
- μετά πρέπει να γίνει μια προσθήκη του χαρακτήρα **T** στη 14^η θέση του **x**
- έτσι, η απόσταση SED μεταξύ **x** και **y** είναι 2.



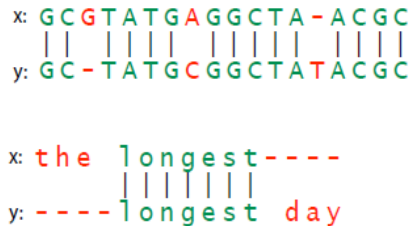
Εικόνα 1 – Παράδειγμα 1

Παράδειγμα 2:

Έστω **x = "GCGTATGAGGCTAACGC"** και **y = "GCTATGCGGCTATACGC"**. Τότε η απόσταση SED μεταξύ **x** και **y** είναι 3, καθώς χρειάζεται μια διαγραφή, μια αλλαγή και μια προσθήκη για να μετατραπεί το **x** σε **y**, όπως φαίνεται και στην εικόνα 2.

Παράδειγμα 3:

Έστω **x = "the longest"** και **y = "longest day"**. Η απόσταση SED μεταξύ **x** και **y** είναι 8, καθώς χρειάζονται 4 διαγραφές και 4 προσθήκες για να μετατραπεί το **x** σε **y**, όπως φαίνεται και στην εικόνα 2.



Εικόνα 2 Παράδειγμα 2 και 3

Υποδειγματική Εκτέλεση

Για να δοκιμάσετε το πρόγραμμα σας μπορείτε να ανακτήσετε από την ιστοσελίδα του μαθήματος το αρχείο **as2-supplementary.zip**, το οποίο περιέχει στον κατάλογο 2.2-supplementary διάφορα δοκιμαστικά αρχεία εισόδου.

Τα ονόματα των αρχείων που περιέχουν τις συμβολοσειρές ορίζονται στη γραμμή εντολών και όλα τα αποτελέσματα εκτυπώνονται στην οθόνη.

```
$./sedcmp dna1.txt dna2.txt
```

```
Minimum edits required to convert is 2
```

```
$./sedcmp dna3.txt dna4.txt
```

```
Minimum edits required to convert is 3
```

```
$./sedcmp ex1.txt ex2.txt
```

Minimum edits required to convert is 8

```
$./sedcmp dnaSequence.txt patSequence.txt
```

Minimum edits required to convert is 43

```
$./sedcmp patSequence.txt dnaSequence.txt
```

Minimum edits required to convert is 43

V. Γενικές Οδηγίες

Το πρόγραμμα σας θα πρέπει να συμβαδίζει με το πρότυπο ISO C, να περιλαμβάνει εύστοχα και περιεκτικά σχόλια, να έχει καλή στοίχιση και το όνομα κάθε μεταβλητής, σταθεράς, ή συνάρτησης να είναι ενδεικτικό του ρόλου της. **Να χρησιμοποιήσετε το λογισμικό τεκμηρίωσης doxygen** έτσι ώστε να μπορούμε να μετατρέψουμε τα σχόλια κάθε ενός από τα προγράμματα σας σε HTML αρχεία και να τα δούμε με ένα browser. Η συστηματική αντιμετώπιση της λύσης ενός προβλήματος περιλαμβάνει τη διάσπαση του προβλήματος σε μικρότερα ανεξάρτητα προβλήματα που κατά κανόνα κωδικοποιούμε σε ξεχωριστές συναρτήσεις. Για αυτό τον λόγο σας καλούμε να κάνετε χρήση συναρτήσεων και άλλων τεχνικών δομημένου προγραμματισμού. Επίσης, σας θυμίζουμε ότι κατά την διάρκεια της εκτέλεσης του προγράμματος σας αυτό θα πρέπει να δίνει τα κατάλληλα μηνύματα λάθος. Τέλος το πρόγραμμα σας θα πρέπει να μεταγλωττίζεται στις μηχανές του εργαστηρίου.

Επίσης να ακολουθήσετε τα πιο κάτω βήματα όταν υποβάλετε την άσκηση σας στο Moodle:

1. Δημιουργείτε ένα κατάλογο με το όνομά σας π.χ., PyrrrosBratskas/ χωρίς να αφήνετε κενά στο όνομα του καταλόγου.
2. Βάλτε μέσα στον κατάλογο αυτό όλα τα αρχεία της εργασίας (κώδικας, doxygen configuration file, README.md) που πρέπει να υποβάλετε.
3. Συμπίεστε (zip) τον κατάλογο (και όχι τα αρχεία ξεχωριστά) χρησιμοποιώντας την εντολή `zip -c PyrrrosBratskas.zip PyrrrosBratskas/*`
4. **Βεβαιωθείτε ότι κάνατε σωστά τα τρία προηγούμενα βήματα**
5. Ανεβάστε στο Moodle το συμπιεσμένο αρχείο π.χ., PyrrrosBratskas.zip

VI. Κριτήρια αξιολόγησης:

Θέμα 1: Πρόγραμμα Σύγκρισης Αρχείων	20
Θέμα 2: Υπολογισμός Αλγόριθμου PageRank	30
- Κριτήριο 1: Σάρωση Αρχείων εισόδου 10	
- Κριτήριο 2: Υπολογισμός & Εκτύπωση Δομών Αλγόριθμου 10	
- Κριτήριο 3: Υπολογισμός Pagerank/Ευκλείδειας & Εκτύπωση 10	
Θέμα 3: (Αλγόριθμος SED)	30
Γενική εικόνα, στοιχισμένος και ευανάγνωστος κώδικας, εύστοχα ονόματα μεταβλητών, σταθερών και συναρτήσεων, σχολιασμός με doxygen).	20
ΣΥΝΟΛΟ	100

Καλή Επιτυχία!