

Πανεπιστήμιο Πατρών

Τμήμα Μηχ. Η/Υ & Πληροφορικής

Ψηφιακές τηλεπικοινωνίες

1η εργαστηριακή άσκηση

Όνομα: Σπύρος Καφτάνης

ΑΜ: 5542

Ερώτημα 1 – Κωδικοποίηση Huffman

Στην κωδικοποίηση Huffman, η απεικόνιση των χαρακτήρων της εισόδου γίνεται σε μεταβλητού μήκους μπλοκ δυαδικών ψηφίων. Η ιδέα του αλγορίθμου αυτού, είναι ότι οι συχνότερα εμφανιζόμενες ακολουθίες απεικονίζονται σε λιγότερα bit, ενώ οι ακολουθίες που εμφανίζονται λιγότερο συχνά απεικονίζονται με περισσότερα. Η κωδικοποίηση αυτή εξασφαλίζει τη μοναδικότητα του διαχωρισμού της λαμβανόμενης δυαδικής ακολουθίας σε κωδικές λέξεις.

Με λίγα λόγια ο αλγόριθμος Huffman:

- i. Διατάζει τις εξόδους της πηγής σε φθίνουσα σειρά των πιθανοτήτων του.
- ii. Συγχωνεύει τις δύο λιγότερο πιθανές εξόδους σε μία μοναδική έξοδο, με τη νέα πιθανότητα να είναι το άθροισμα των δύο πιθανοτήτων μέχρι να έχει απομείνει μόνο ένα στοιχείο με πιθανότητα 1.
- iii. Αυθαίρετη αντιστοίχιση των τιμών 0 και 1 ως κωδικές λέξεις. (πχ. Αν φανταστούμε την αναπαράσταση σε μία δομή δέντρου μπορούμε να πούμε ότι όσα βρίσκονται σε δεξί κλάδο θα πάρουν τη τιμή 1, ενώ όποια βρίσκονται σε αριστερό τη τιμή 0).

1. Στο συγκεκριμένο ερώτημα υλοποιήθηκαν οι συναρτήσεις: myHuffmandict, myHuffmanenco, myHuffmandeco αντίστοιχες των huffmandict, huffmanenco, huffmandeco.

Πιο αναλυτικά:

- **function** [symbols, representation, avglen] = myHuffmandict (alphabet, p)
Παίρνει σαν όρισμα ένα αλφάβητο (alphabet) και τις αντίστοιχες πιθανότητες του κάθε συμβόλου (p) και επιστρέφει ένα διάνυσμα με τα σύμβολα (symbols) και την αντίστοιχη αναπαράσταση του κάθε συμβόλου (representation). Τέλος επιστρέφει το μέσο μήκος κώδικα (avglen). Για να βρούμε την αναπαράσταση κάποιου συμβόλου X θα δώσουμε την εντολή: representation(find(symbols==X)) (βλ. MyHuffmanenco).
- **function** [comp] = myHuffmanenco (actualsig, representation, symbols)
Η συνάρτηση αυτή παίρνει σαν όρισμα το "κείμενο" (actualsig), την αναπαράσταση (representation) και τα σύμβολα (symbols) που αντιστοιχούν 1 προς 1 στην αναπαράσταση. Επιστρέφει το κωδικοποιημένο μήνυμα (comp).
- **function** [dsig] = myhuffmandeco(comp, representation, symbols)
Η συνάρτηση αυτή παίρνει σαν όρισμα το κωδικοποιημένο μήνυμα (comp), τα

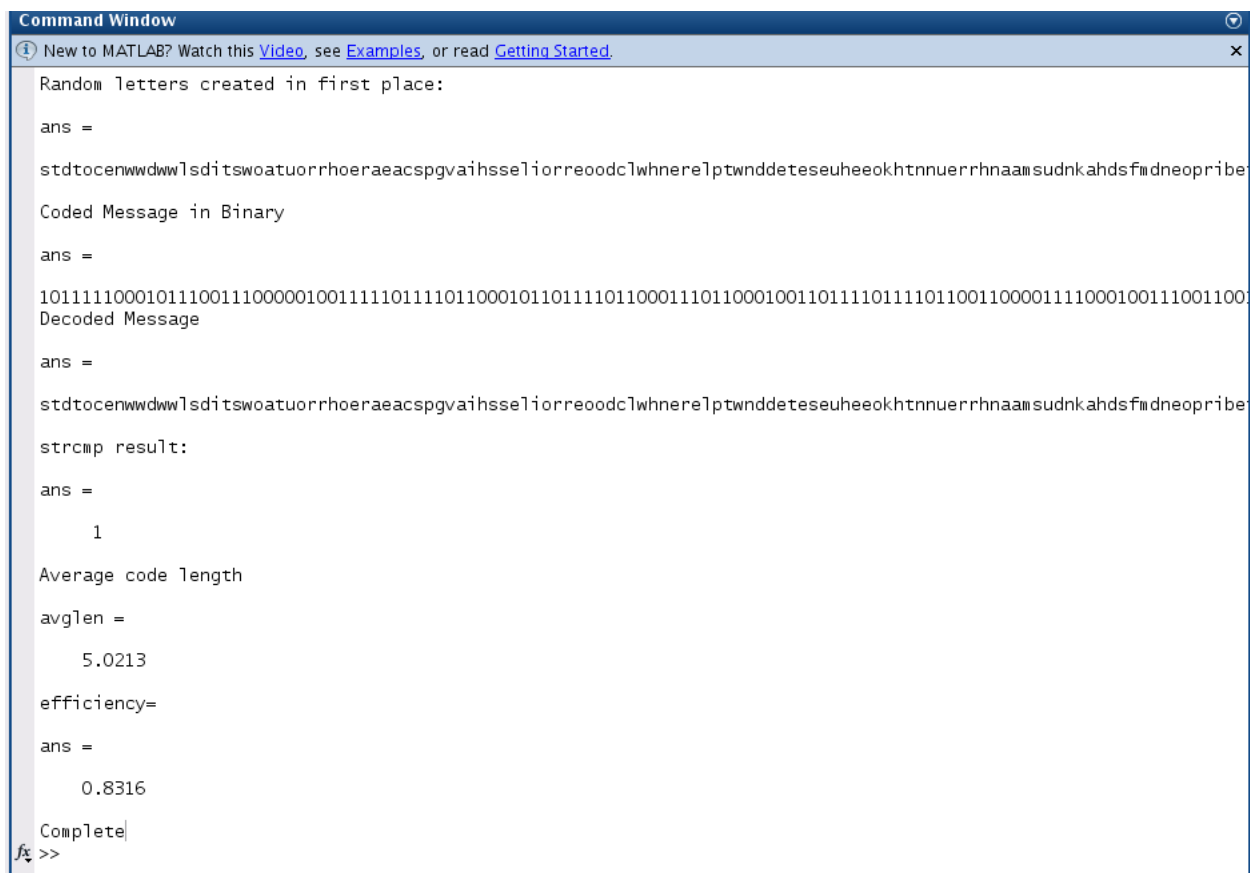
σύμβολα του αλφαβήτου (symbols) και την αντίστοιχη αναπαράσταση (representation) και επιστρέφει το αποκωδικοποιημένο μήνυμα.

Ο κώδικας του ερωτήματος βρίσκεται στα αρχεία myHuffmandict.m, myHuffmanenco.m ,myHuffmandeco.m

2.

Ο κώδικας του συγκεκριμένου ερωτήματος βρίσκεται στο αρχείο erotima2A.m για την πηγή A και στο erotima2B.m για την πηγή B.

Για την πηγή A, δημιουργούμε μια τυχαία έξοδο 10000 χαρακτήρων (1-26) οι οποίες αντιστοιχούν στα γράμματα τις αλφαβήτας. Μετά την εκτέλεση του erotima2A.m μπορούμε να επιβεβαιώσουμε την ορθή αποκωδικοποίηση συγκρίνοντας το dsig, που είναι το αποκωδικοποιημένο μήνυμα με το out, που ήταν το αρχικό, το οποίο παράχθηκε τυχαία. Στην εκτέλεση του script εμφανίζεται το αποτέλεσμα της σύγκρισης αυτών των strings καθώς και το μέσο μήκος κώδικα το οποίο είναι περίπου 5.



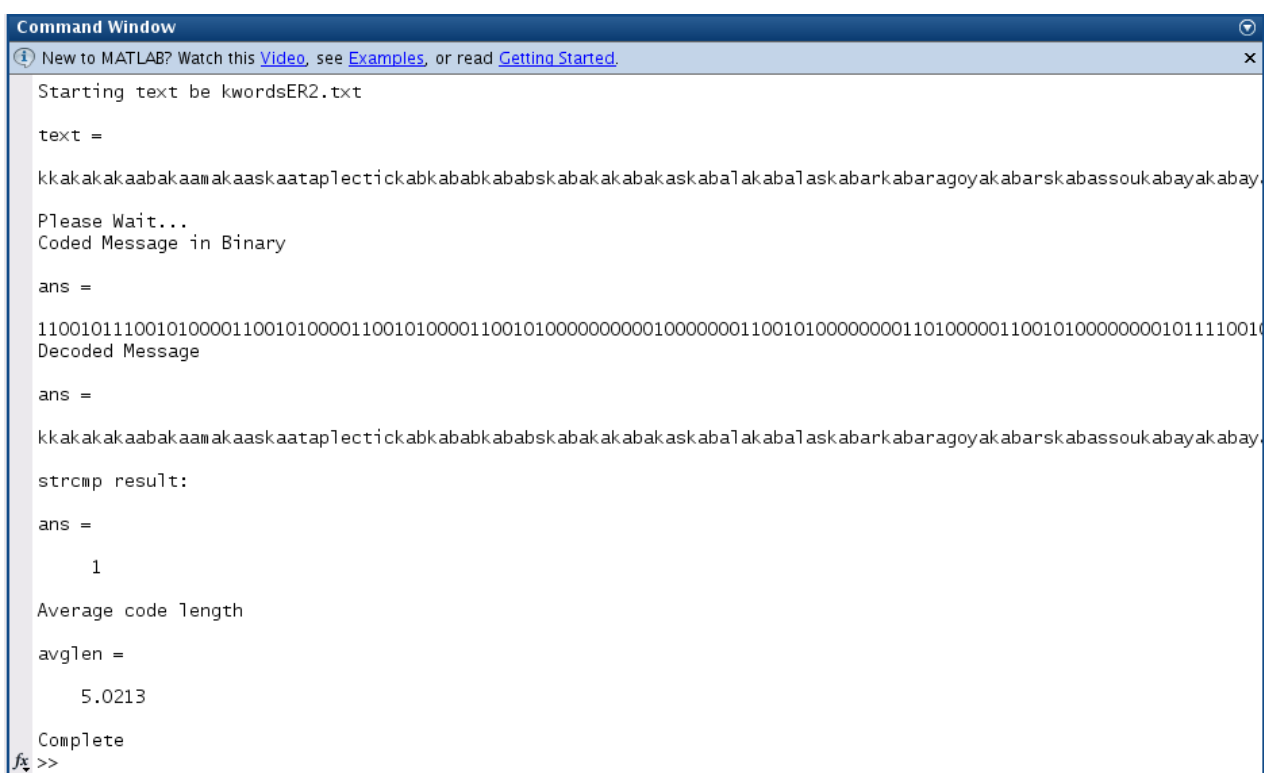
```
Command Window
New to MATLAB? Watch this Video, see Examples, or read Getting Started.
Random letters created in first place:
ans =
stdtocenwwdwwlsditswoatuorrhoeraeacspgvaihsseliorreoodclwhnereiptwddeteseuheekhtnnuerrhnaamsudnkaahdsfmdneopribe
Coded Message in Binary
ans =
10111110001011100111000001001111101111011000101101111011000111011000100110111101111011001100001111000100111001100
Decoded Message
ans =
stdtocenwwdwwlsditswoatuorrhoeraeacspgvaihsseliorreoodclwhnereiptwddeteseuheekhtnnuerrhnaamsudnkaahdsfmdneopribe
strcmp result:
ans =
1
Average code length
avglen =
5.0213
efficiency=
ans =
0.8316
Complete
fx >>
```

Εικόνα 1: Εκτέλεση erotima2A.m

Για το ερώτημα αυτό έχουμε δημιουργήσει δύο βοηθητικές συναρτήσεις, οι οποίες χειρίζονται τη μετατροπή από γράμματα σε αλφάβητο και αντίστροφα. Συγκεκριμένα η `convertVector2Letters` μετατρέπει ένα διάνυσμα αριθμών σε γράμματα (το 1 είναι το a, το 2 το b κτλ), ενώ η `convert2numbers` κάνει την αντίστροφη διαδικασία. Επίσης υπάρχουν και άλλες δύο με παρόμοιες λειτουργίες. Οι διαφορές τους εντοπίζεται στο αν τα δεδομένα εισόδου/εξόδου είναι σε `string` ή σε διάνυσμα.

(Σημείωση: Το script χρειάζεται κάποια δευτερόλεπτα για να ολοκληρώσει τους υπολογισμούς).

Για την πηγή Β, όπως διευκρινίστηκε στο forum, χρησιμοποιούμε το ίδιο αλφάβητο με την πηγή Α, δηλαδή δε χρησιμοποιούμε μόνο πεζά αγγλικά γράμματα. Αφού αφαιρέθηκαν οι περιττοί χαρακτήρες και έγιναν οι κατάλληλες αλλαγές δημιουργήθηκε το αρχείο `keywordsER2.txt` το οποίο και χρησιμοποιήθηκε εδώ. Στο script `erotima2B`, χρησιμοποιήθηκαν οι κατάλληλες συναρτήσεις για τις μετατροπές και η διαδικασία εκτελέστηκε με επιτυχία.



```
Command Window
New to MATLAB? Watch this Video, see Examples, or read Getting Started.
Starting text be keywordsER2.txt

text =

kkakakakaabakaamakaaskaataplectickabkababkababskabakakabakaskaba1akaba1askabarkabaragoyakabarskabassoukabayakabay.

Please Wait...
Coded Message in Binary

ans =

11001011100101000011001010000110010100001100101000000000100000001100101000000001101000001100101000000001011110010

Decoded Message

ans =

kkakakakaabakaamakaaskaataplectickabkababkababskabakakabakaskaba1akaba1askabarkabaragoyakabarskabassoukabayakabay.

strcmp result:

ans =

    1

Average code length

avglen =

    5.0213

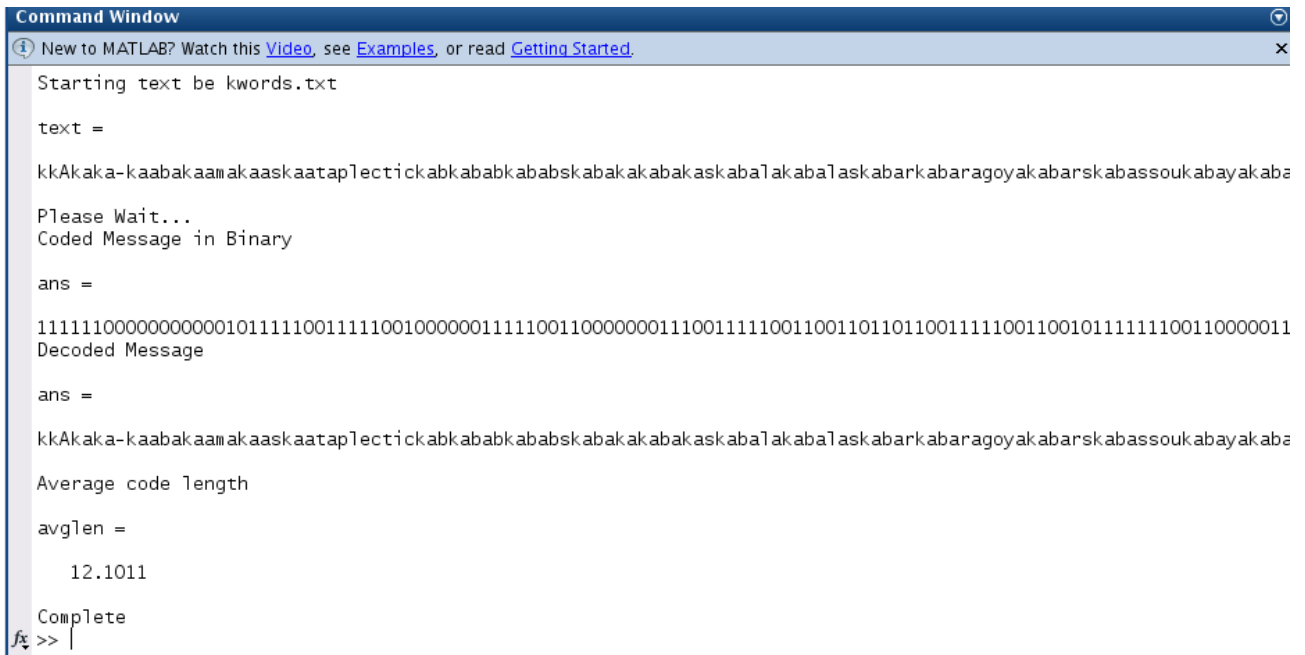
Complete
fx >>
```

Εικόνα 2: Εκτέλεση `erotima2B.m`

Το μέσο μήκος κώδικα είναι ίδιο με τη πρώτη πηγή, μιας και το αλφάβητο και οι πιθανότητες εμφάνισης παραμένουν ίδιες.

3.

Στο ερώτημα αυτό χρησιμοποιήθηκε το αρχείο kwords.txt. Βρέθηκε το νέο αλφάβητο με τη βοήθεια της unique και υπολογίσθηκαν οι νέες πιθανότητες εμφάνισής του κάθε χαρακτήρα στο αρχείο. Η υλοποίηση βρίσκεται στο αρχείο erotima3.m. Η εκτέλεσή του δίνει τη παρακάτω έξοδο.



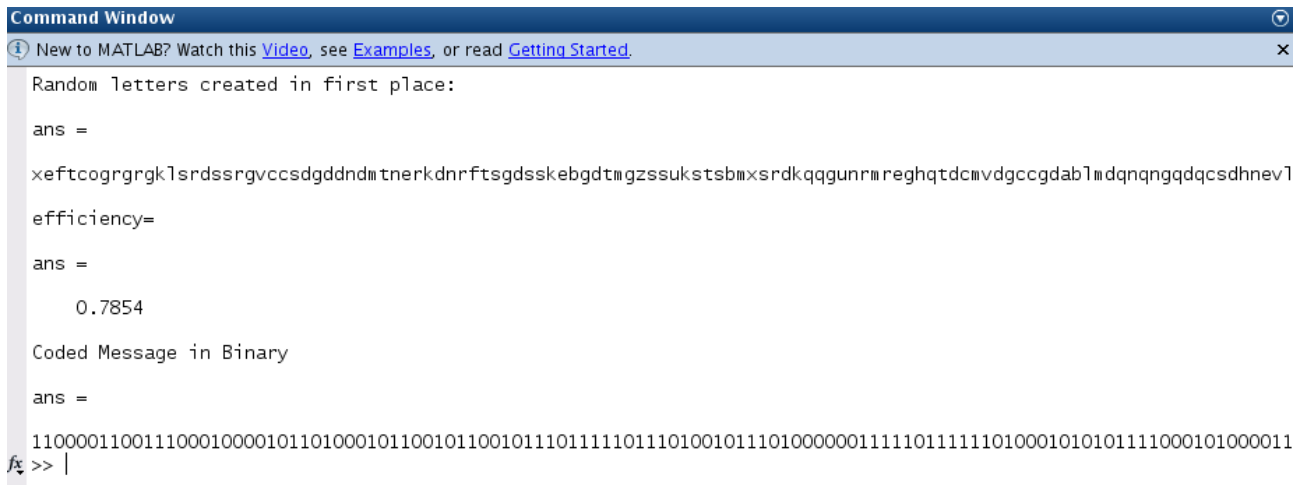
```
Command Window
New to MATLAB? Watch this Video, see Examples, or read Getting Started.
Starting text be kwords.txt
text =
kkAkaka-kaabakaamakaaskaataplectickabkababkababskabakakabakaskabaalakabaskabarkabaragoyakabarskabassoukabayakaba
Please Wait...
Coded Message in Binary
ans =
1111110000000000010111110011111001000000111110011000000011100111110011001101101100111110011001011111100110000011
Decoded Message
ans =
kkAkaka-kaabakaamakaaskaataplectickabkababkababskabakakabakaskabaalakabaskabarkabaragoyakabarskabassoukabayakaba
Average code length
avglen =
12.1011
Complete
fx >> |
```

Εικόνα 3: Εκτέλεση erotima3.m

Το μήκος κωδικοποίησης είναι μεγαλύτερο κατά 7 μονάδες περίπου, μιας και το νέο αλφάβητο είναι μεγαλύτερο.

4.

Στο ερώτημα αυτό υπολογίστηκε η δεύτερη τάξης επέκταση της πηγής A. Η πιθανότητα του κάθε ζεύγους προέκυψε από το γινόμενο των επί μέρους πιθανοτήτων. Τα αποτελέσματα που προκύπτουν από μία εκτέλεση του script erotima4.m φαίνονται στη παρακάτω εικόνα.



```
Command Window
New to MATLAB? Watch this Video, see Examples, or read Getting Started.
Random letters created in first place:
ans =
xftcogrgrgklsrdssrgvccsdgddndmtnerkdnrftsgdsskebgdtmgzssukstsbmxsrkqggunrmreghqtdcmvdgccgdablm dqnqngqdqcsdhnev1
efficiency=
ans =
0.7854
Coded Message in Binary
ans =
110000110011100010000101101000101100101100101110111101110100101110100000011111011111010001010101111000101000011
fx >> |
```

Εικόνα 4: Εκτέλεση erotima4.m

Όπως φαίνεται, έχει υπολογιστεί η αποδοτικότητα (efficiency) που προκύπτει από το γινόμενο της εντροπίας με το μέσο μήκος λέξης. Η τιμή αυτή είναι 0.7854, τη στιγμή που στο ερώτημα 2 με τη χρήση πάλι της πηγής A, αλλά χωρίς τη χρήση της δεύτερης τάξης επέκτασης ήταν 0.8316 (εικόνα 1). Το γεγονός αυτό αποδεικνύει τη θεωρία, δηλαδή ότι όσο ανεβαίνουμε σε τάξη επέκτασης τόσο καλύτερη αποδοτικότητα πετυχαίνουμε και τόσο πιο κοντά θα φτάνουμε στο 1.

Αρχικά, η κωδικοποίηση της πηγής B γίνεται με τις πιθανότητες των ζευγών του προηγούμενου ερωτήματος. Επειδή το προηγούμενο ερώτημα αφορούσε τη πηγή A, αφαιρούμε τα επιπλέον σύμβολα, δηλαδή χρησιμοποιούμε το αρχείο kwordsER2.txt.

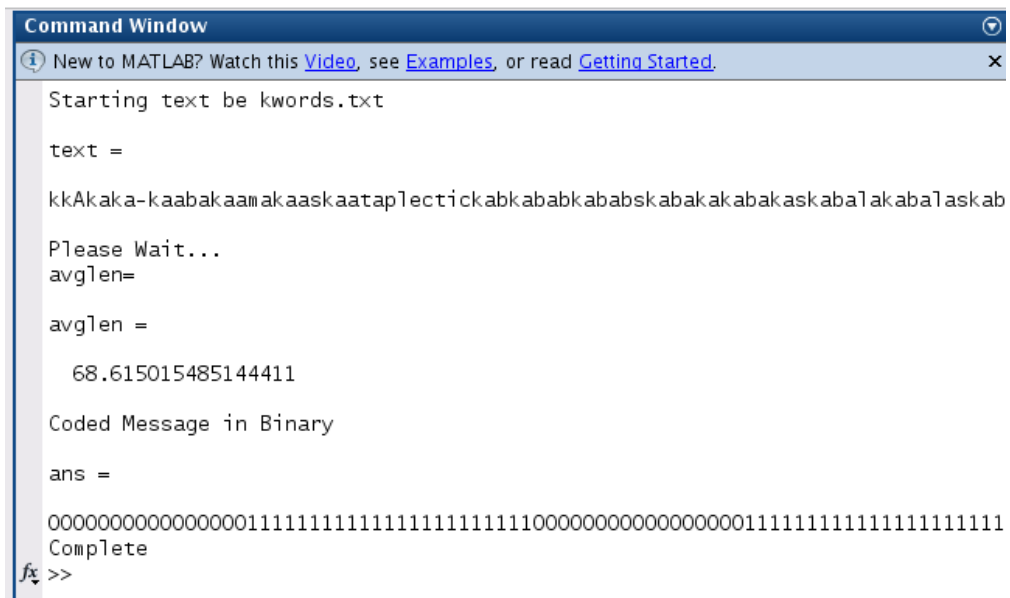
Η έξοδος του script φαίνεται στην εικόνα 5.

[illegible]

Εικόνα 5: Εκτέλεση erotima 5A.m

Στην συνέχεια υπολογίσθηκαν οι νέες πιθανότητες οι οποίες προέκυψαν από τη συχνότητα εμφάνισης του κάθε συνδυασμού στο αρχείο keywords.txt. Η εκτέλεση του erotima5B.m έδωσε τα παρακάτω αποτελέσματα.

Το μέσο μήκος κώδικα είναι μεγαλύτερο σε σχέση με την εκτέλεση της πηγής B ως πρώτης τάξης, επειδή οι πιθανότητες εμφάνισης των συνδυασμών είναι πολύ μικρές.



```
Command Window
New to MATLAB? Watch this Video, see Examples, or read Getting Started.
Starting text be kwords.txt
text =
kkAkaka-kaabakaamakaaskaataplectickabkababkababskabakakabakaskabalakabaaskab
Please Wait...
avglen=
avglen =
    68.615015485144411
Coded Message in Binary
ans =
00000000000000001111111111111111110000000000000000111111111111111111
Complete
fx >>
```

Εικόνα 6: εκτέλεση του erotima5B.m

Κώδικας

```
function [ symbols, representation, avglen ] = myHuffmandict (alphabet, p)

%lengthx2Sorted: Ταξινομημένο κατά φθίνουσα σειρά των πιθανοτήτων
Alen=length(alphabet);
IDs=[1:Alen];
C=cell(length(alphabet),2);
for i=1:length(alphabet)
    C{i}=p(i);
    C{i,2}=alphabet(i);
end

avglen=0;

%φθίνουσα ταξινόμηση με βάση τις πιθανότητες.
Sorted=sortrows(C,[-1,1]);
StartSorted=Sorted;

%το διάνυσμα αυτό περιλαμβάνει τον προσωρινό κώδικα την κάθε λέξης
%(παράλληλο διάνυσμα με το ταξινομημένο μητρώο Sorted.
%Αρχική τιμή ==-1 -> δεν έχει προσπελαθεί ακόμα
for i=1:Alen Coding{i}=""; end

for i=Alen:-1:2

    if Sorted{i,2}(1) > Sorted{i-1,2}(1)
        for j=1:length(Sorted{i,2})
            Coding{Sorted{i,2}(j)}=['1' Coding{Sorted{i,2}(j)}];
        end
        for j=1:length(Sorted{i-1,2})
            Coding{Sorted{i-1,2}(j)}=['0' Coding{Sorted{i-1,2}(j)}];
        end
    else
        for j=1:length(Sorted{i,2})
            Coding{Sorted{i,2}(j)}=['0' Coding{Sorted{i,2}(j)}];
        end
        for j=1:length(Sorted{i-1,2})
            Coding{Sorted{i-1,2}(j)}=['1' Coding{Sorted{i-1,2}(j)}];
        end
    end
end
```

```

end

newProb=Sorted{i,1}+Sorted{i-1,1}; %άθροισμα των δύο μικρότερων πιθανοτήτων
Sorted{i,1}=0; %μηδενισμός τελευταίας γραμμής (αφαίρεση)
Sorted{i-1}=newProb; %το προηγούμενο άθροισμα στέκεται σαν μια νέα πιθανότητα
στον πίνακα
Sorted{i-1,2}=[Sorted{i-1,2} Sorted{i,2}];

Sorted=sortrows(Sorted, [-1,1]); %εκ νέου ταξινόμηση

%ταξινόμηση των cells (για σωστή διαχείριση των 0 και 1
for j=1:Alen
    Sorted{j,2}=sort(Sorted{j,2});
end

end

representation=Coding';
symbols=[1:Alen]';

%υπολογισμός του μέσου μήκους κώδικα
sum=0;
for i=1:Alen
    sum=sum+numel(Coding{i})*StartSorted{i,1}; %Σ(p*I)
end
avglen=sum;

end

```

```

function [ dsig ] = myhuffmandeco( comp, representation, symbols )

chars="";
dsig="";
tempValue=0;
exist=0;
counter=1;
while 1
    if counter > length(comp{1})
        dsig=strcat(dsig,num2str(tempValue));
        dsig=strcat(dsig, '|');
        break
    end
end

```

```

chars=strcat(chars, comp{1}(counter));
for j=1:length(symbols)
    if length(representation{j}) >= length(chars)
        if strcmp(representation{j}(1:length(chars)), chars)
            exist=1;
        end
    end
end
if exist == 1
    exist=0;
    tempValue=symbols(find(strcmp(representation,chars)));
    counter=counter+1;
else
    dsig=strcat(dsig,num2str(tempValue));
    dsig=strcat(dsig, '|');
    chars="";
end

end

end

```

```

function [ comp ] = myHuffmanenco ( actualsig, representation, symbols )

%σε κάθε επανάληψη στο comp προστίθεται η κωδικοποίηση του κάθε συμβόλου
%Στο τέλος προκύπτει η αναπαράσταση όλου του κειμένου
comp="";
for i=1:length(actualsig)
    comp=strcat(comp,representation(find(symbols==actualsig(i))));
end

end

```

%Ερώτημα 2Α (σε μορφή script)

```

clear
clc
%δημιουργία αλφαβήτου. Οι αριθμοί 1-26 αντιστοιχούν στα 26 γράμματα του

```

```

%αλφαβήτου
alphabet=[1:26];

%οι πιθανότητες που αντιστοιχούν στο κάθε γράμμα του αλφαβήτου
p=[8.167/100,1.492/100,2.782/100,4.253/100,12.702/100,2.228/100,2.015/100,6.094/100,6.966/100,0.153/100,0.772/100,4.025/100,2.406/100,6.749/100,7.507/100,1.929/100,0.095/100,5.987/100,6.327/100,9.056/100,2.758/100,0.978/100,2.361/100,0.150/100,1.974/100,0.074/100];

%τυχαία πηγή 10000ων συμβόλων από το συγκεκριμένο αλφάβητο.
out = randsrc(1,10000,[alphabet; p]);
disp('Random letters created in first place:');
convertVector2Letters(out)

%δημιουργία αναπαράστασης
[sym, repr, avglen] = myHuffmandict (alphabet, p);

%κωδικοποίηση
[comp] = myHuffmanenco (out, repr, sym);
disp('Coded Message in Binary');
comp{1}

%αποκωδικοποίηση
[dsig] = myhuffmandeco(comp, repr, sym);
disp('Decoded Message');
convert2letters(dsig)

disp('strcmp result:');
strcmp(convertVector2Letters(out), convert2letters(dsig))

disp('Average code length');
avglen

entropy=0;
%υπολογισμός εντροπίας
for i=1:length(p)
    entropy=entropy+p(i)*log2(1./p(i));
end

disp('efficiency=');

```

```
entropy./avglen
```

```
disp('Complete')
```

```
%Ερώτημα 2B (σε μορφή script)
```

```
clear
```

```
clc
```

```
%δημιουργία αλφαβήτου. Οι αριθμοί 1-26 αντιστοιχούν στα 26 γράμματα του
```

```
%αλφαβήτου
```

```
alphabet=[1:26];
```

```
%οι πιθανότητες που αντιστοιχούν στο κάθε γράμμα του αλφαβήτου
```

```
p=[8.167/100,1.492/100,2.782/100,4.253/100,12.702/100,2.228/100,2.015/100,6.094/100,6.966/100,0.153/100,0.772/100,4.025/100,2.406/100,6.749/100,7.507/100,1.929/100,0.095/100,5.987/100,6.327/100,9.056/100,2.758/100,0.978/100,2.361/100,0.150/100,1.974/100,0.074/100];
```

```
%πηγή που προέρχεται από το αρχείο kwordsER2.txt
```

```
text = fileread('kwordsER2.txt');
```

```
text = strrep(text,sprintf('\n'),"); %αφαίρεση new lines
```

```
disp('Starting text be kwordsER2.txt')
```

```
text
```

```
disp('Please Wait...');
```

```
tempout=convert2numbers(text); %μετατροπή σε αριθμούς
```

```
%μετατροπή σε πίνακα αριθμών (out)
```

```
out=convert2NumberVector(tempout, 29096);
```

```
out=out';
```

```
%δημιουργία αναπαράστασης
```

```
[sym, repr, avglen] = myHuffmandict (alphabet, p);
```

```
%κωδικοποίηση
```

```
[comp] = myHuffmanenco (out, repr, sym);
```

```
disp('Coded Message in Binary');
```

```
comp{1}
```

```
%αποκωδικοποίηση
```

```

[dsig] = myhuffmandeco(comp, repr, sym);
disp('Decoded Message');
convert2letters(dsig)

disp('strcmp result:');
strcmp(convertVector2Letters(out), convert2letters(dsig))

disp('Average code length');
avglen

disp('Complete')

```

%Ερώτημα 3 (σε μορφή script)

```

clear
clc

text = fileread('kwords.txt');
text = strrep(text,sprintf("\n"),"); %αφαίρεση new lines
disp('Starting text be kwords.txt')
text

%εύρεση νέων πιθανοτήτων εμφάνισης
%για κάθε μοναδικό σύμβολο, πόσες φορές έχει εμφανιστεί

diffSymbols=unique(text);
p=zeros(length(diffSymbols),1);

for i=1:length(diffSymbols)
    p(i)=length(strfind(text,diffSymbols(i)))/length(text);
end
p=p';

alphabet=[1:length(diffSymbols)];

disp('Please Wait...');

tempout=convert2numbersB(text,diffSymbols); %μετατροπή σε αριθμούς
%μετατροπή σε πίνακα αριθμών (out)
out=convert2NumberVector(tempout, length(text));
out=out';

%δημιουργία αναπαράστασης

```

```
[sym, repr, avglen] = myHuffmandict (alphabet, p);
```

```
%κωδικοποίηση
```

```
[comp] = myHuffmanenco (out, repr, sym);
```

```
disp('Coded Message in Binary');
```

```
comp{1}
```

```
%αποκωδικοποίηση
```

```
[dsig] = myhuffmandeco(comp, repr, sym);
```

```
disp('Decoded Message');
```

```
convert2lettersB(dsig,diffSymbols)
```

```
disp('Average code length');
```

```
avglen
```

```
disp('Complete')
```

```
%Ερώτημα 4 (σε μορφή script)
```

```
clear
```

```
clc
```

```
%δημιουργία αλφαβήτου. Οι αριθμοί 1-26 αντιστοιχούν στα 26 γράμματα του
```

```
%αλφαβήτου
```

```
startAlphabet=[1:26];
```

```
%οι πιθανότητες που αντιστοιχούν στο κάθε γράμμα του αλφαβήτου
```

```
startP=[8.167/100,1.492/100,2.782/100,4.253/100,12.702/100,2.228/100,2.015/100,6.09  
4/100,6.966/100,0.153/100,0.772/100,4.025/100,2.406/100,6.749/100,7.507/100,1.929/  
100,0.095/100,5.987/100,6.327/100,9.056/100,2.758/100,0.978/100,2.361/100,0.150/10  
0,1.974/100,0.074/100];
```

```
%Για τη δεύτερης τάξης επέκταση, θέλουμε όλους τους πιθανούς συνδιασμούς
```

```
%ανά δύο. Αυτό δίνεται από το αποτέλεσμα της combnk συν τα αντίστροφα + το
```

```
%καθένα με τον εαυτό του.
```

```
first=combnk(startAlphabet,2);
```

```
second=flipr(combnk(startAlphabet,2));
```

```
final=[first;second];
```

```

for i=1:26
    final=[final;[i,i]];
end

p=zeros(length(final),1);
%νέες πιθανότητες
for i=1:length(final)
    p(i)=startP(final(i,1))*startP(final(i,2));
end
p=p';

alphabet=[1:length(final)];

%τυχαία πηγή 5000ων συμβόλων από το συγκεκριμένο αλφάβητο.
out = randsrc(1,10000,[alphabet; p]); %10000 χαρακτήρες=5000 ζεύγη
disp('Random letters created in first place:');
convertVector2Letters(out)

%δημιουργία αναπαράστασης
[sym, repr, avglen] = myHuffmandict (alphabet, p);

%κωδικοποίηση
[comp] = myHuffmanenco (out, repr, sym);

entropy=0;
%υπολογισμός εντροπίας
for i=1:length(p)
    entropy=entropy+p(i)*log2(1./p(i));
end

disp('efficiency=');
entropy./avglen

disp('Coded Message in Binary');
comp{1}

disp('Complete')

```


%Ερώτημα 5Α (σε μορφή script)

clear

clc

text = fileread('kwordsER2.txt');

text = strrep(text,sprintf('\n'),"); %αφαίρεση new lines

disp('Starting text be kwords.txt')

text

diffSymbols=unique(text);

p=zeros(length(diffSymbols),1);

%πιθανότητες κάθε χαρακτήρα

startP=[8.167/100,1.492/100,2.782/100,4.253/100,12.702/100,2.228/100,2.015/100,6.094/100,6.966/100,0.153/100,0.772/100,4.025/100,2.406/100,6.749/100,7.507/100,1.929/100,0.095/100,5.987/100,6.327/100,9.056/100,2.758/100,0.978/100,2.361/100,0.150/100,1.974/100,0.074/100];

alphabet=[1:length(diffSymbols)];

disp('Please Wait...');

startAlphabet=alphabet;

%Για τη δεύτερης τάξης επέκταση, θέλουμε όλους τους πιθανούς συνδιασμούς
%ανά δύο. Αυτό δίνεται από το αποτέλεσμα της combnk συν τα αντίστροφα + το
%καθένα με τον εαυτό του.

first=combnk(startAlphabet,2);

second=flipr(combnk(startAlphabet,2));

final=[first;second];

for i=1:26

 final=[final;[i,i]];

end

p=zeros(length(final),1);

%νέες πιθανότητες (ζευγών χαρακτήρων)

for i=1:length(final)

```

    p(i)=startP(final(i,1))*startP(final(i,2));
end
p=p';

alphabet=[1:length(final)];

%μετατροπή σε αριθμούς
tempout=convert2numbersB(text,diffSymbols);
%μετατροπή σε πίνακα αριθμών (out)
out=convert2NumberVector(tempout, length(text));
out=out';

%δημιουργία αναπαράστασης
[sym, repr, avglen] = myHuffmandict (alphabet, p);

%κωδικοποίηση
[comp] = myHuffmanenco (out, repr, sym);

entropy=0;
%υπολογισμός εντροπίας
for i=1:length(p)
    entropy=entropy+p(i)*log2(1./p(i));
end

disp('efficiency=');
entropy./avglen

disp('avglen=');
avglen

disp('Coded Message in Binary');
comp{1}

disp('Complete')

```

```

%Ερώτημα 5B (σε μορφή script)

clear
clc

text = fileread('kwords.txt');
text = strrep(text,sprintf("\n"),"); %αφαίρεση new lines
disp('Starting text be kwords.txt')
text

diffSymbols=unique(text);
p=zeros(length(diffSymbols),1);

alphabet=[1:length(diffSymbols)];

disp('Please Wait...');

startAlphabet=alphabet;

%Για τη δεύτερης τάξης επέκταση, θέλουμε όλους τους πιθανούς συνδιασμούς
%ανά δύο. Αυτό δίνεται από το αποτέλεσμα της combnk συν τα αντίστροφα + το
%καθένα με τον εαυτό του.

first=combnk(startAlphabet,2);
second=fliplr(combnk(startAlphabet,2));
final=[first;second];
for i=1:26
    final=[final;[i,i]];
end

p=zeros(length(final),1);

%νέες πιθανότητες (ζευγών χαρακτήρων) υπολογισμένες από το αρχείο
*****
for i=1:length(final)
    %πόσες φορές υπάρχει στο text ο κάθε συνδιασμός / όλους τους
    %συνδιασμούς (στη σειρά)
    p(i)=(length(strfind(text,strcat(diffSymbols(final(i,1)),diffSymbols(final(i,2)))))/
(length(text)-1);

```

```

end
p=p';

alphabet=[1:length(final)];

%μετατροπή σε αριθμούς
tempout=convert2numbersB(text,diffSymbols);
%μετατροπή σε πίνακα αριθμών (out)
out=convert2NumberVector(tempout, length(text));
out=out';

%δημιουργία αναπαράστασης
[sym, repr, avglen] = myHuffmandict (alphabet, p);

%κωδικοποίηση
[comp] = myHuffmanenco (out, repr, sym);

disp('avglen=');
avglen

disp('Coded Message in Binary');
comp{1}

disp('Complete')

```

Βοηθητικές συναρτήσεις

```
function [ letters ] = convert2letters ( numbers )

realAlphabet=['a':'z'];
realNumbers=[1:26];
letters="";
chars=""; %temp
for i=1:length(numbers)
    if numbers(i) == 'I'
        letters=strcat(letters,realAlphabet(find(realNumbers==numbers(i))));
        letters=strcat(letters,realAlphabet(str2num(chars)));
        chars="";
    else
        chars=strcat(chars,numbers(i));
    end
end
end
```

```
function [ letters ] = convert2lettersB ( numbers, realAlphabet )

realNumbers=[1:length(realAlphabet)];
letters="";
chars=""; %temp
for i=1:length(numbers)
    if numbers(i) == 'I'
        letters=strcat(letters,realAlphabet(find(realNumbers==numbers(i))));
        letters=strcat(letters,realAlphabet(str2num(chars)));
        chars="";
    else
        chars=strcat(chars,numbers(i));
    end
end
end
```

```

function [ numbers ] = convert2numbers( letters )

    realAlphabet=['a':'z'];

    numbers="";
    for i=1:length(letters)
        numbers=strcat(numbers, num2str(find(realAlphabet==letters(i)))) );
        numbers=strcat(numbers, '|'); %διαχωριστικό για να ξέρουμε που αλλάξει γράμμα
    end

end

```

```

function [ numbers ] = convert2numbersB ( letters, realAlphabet)

    numbers="";
    for i=1:length(letters)
        numbers=strcat(numbers, num2str(find(realAlphabet==letters(i)))) );
        numbers=strcat(numbers, '|'); %διαχωριστικό για να ξέρουμε που αλλάξει γράμμα
    end

end

```

```

function [ vector ] = convert2NumberVector( numbers, textLength )

    chars="";
    counter=1;
    pipe=0; %for a bug
    vector=zeros(textLength,1);
    for i=1:length(numbers)
        if numbers(i) == '|'
            if pipe == 0
                vector(counter)=str2num(chars);
                counter=counter+1;
                chars="";
                pipe=1;
            end
        else
            chars=strcat(chars,numbers(i));
            pipe=0;
        end
    end

```

```
end
```

```
end
```

```
function [ letters ] = convertVector2Letters( numbers )
```

```
    realAlphabet=['a':'z'];
```

```
    realNumbers=[1:26];
```

```
    letters="";
```

```
    for i=1:length(numbers)
```

```
        letters=strcat(letters, realAlphabet(find(realNumbers==numbers(i))));
```

```
    end
```

```
end
```