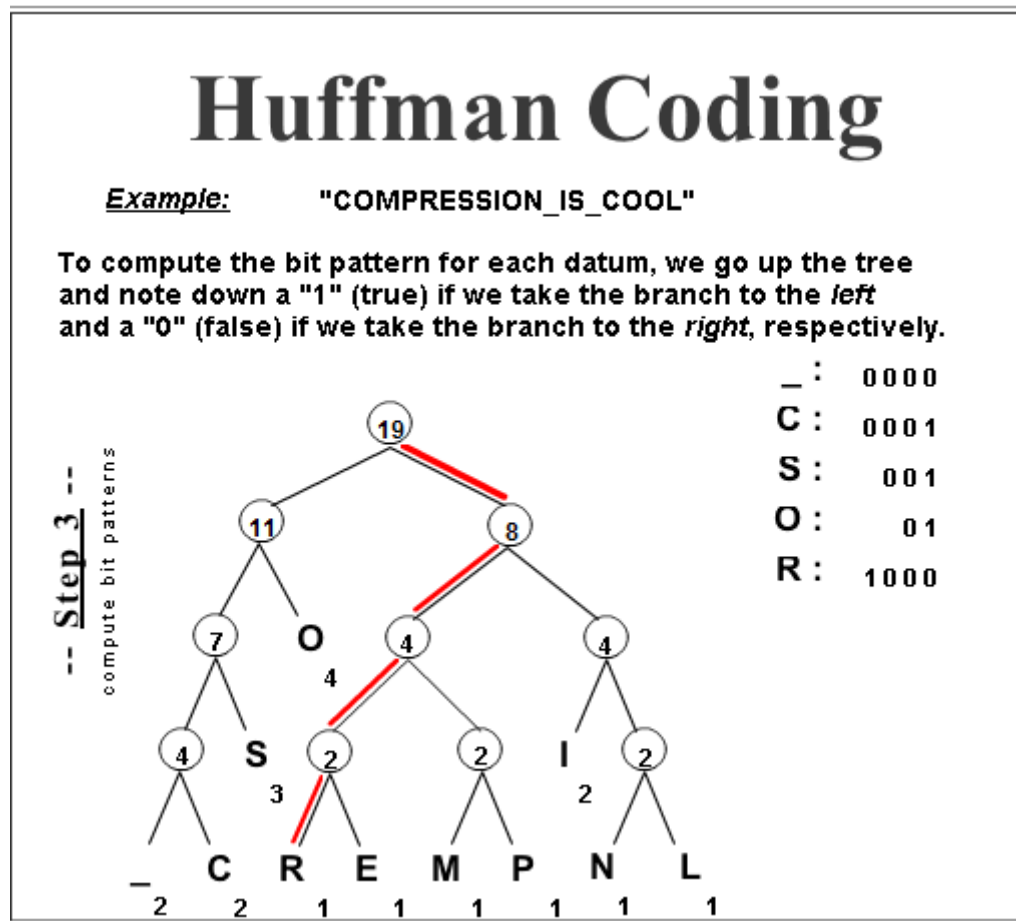


ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

Ψηφιακές Τηλεπικοινωνίες



1^ο Ερώτημα

ΑΚΑΔΗΜΑΙΚΟ ΈΤΟΣ 2015-2016

ΜΑΛΛΙΟΣ ΧΑΡΑΛΑΜΠΟΣ 5343

Ερώτημα 1 – κωδικοποίηση Huffman

Γενικά, αν έχουμε ένα σύνολο από n χαρακτήρες s_1, s_2, \dots, s_n , που ο κάθε ένας συμβαίνει με πιθανότητα $P(s_i)$, τότε:

$$H = -\sum_{i=1}^n P(s_i) \log_2 P(s_i)$$

Τα βήματα της μεθόδου αυτής, η οποία προτάθηκε από τον Huffman το 1958, είναι τα εξής

1. Διατάσσουμε τα σύμβολα της πηγής σε φθίνουσα διάταξη σύμφωνα με τις πιθανότητές τους.
2. Αντικαθιστούμε τα δύο τελευταία σύμβολα με ένα σύμβολο του οποίου δείκτης είναι οι δείκτες των συμβόλων που αντικαθιστά ενώ η πιθανότητά του είναι το άθροισμα των πιθανοτήτων των συμβόλων από τα οποία προέκυψε.
3. Στα σύμβολα που αντικαταστάθηκαν αντιστοιχούμε τους κωδικούς χαρακτήρες 0 και 1 ή 1 και 0.
4. Επαναλαμβάνουμε τη διαδικασία που περιγράψαμε. Η κωδική λέξη κάθε συμβόλου προκύπτει διαβάζοντας από τα δεξιά προς κάθε ένα σύμβολο, παίρνοντας τον αντίστοιχο κωδικό χαρακτήρα όπου υπάρχει δείκτης του κωδικοποιούμενου συμβόλου.

Ζητούμενο 1

α)

*Δημιουργία της συνάρτησης **huffmandict** : δημιουργεί κωδικές λέξεις('kw dik') χρησιμοποιώντας το αλφάβητο εισόδου('alphabito') και τις δοσμένες πιθανότητες('pithanotites').*

Κώδικας:

```
function [ kw dik ] = huffmandict(alphabito,pithanotites)

%prokeimenou na tairiazoun oi diastaseis kanw reshape sto yparxon kai
to
%anathetw sto alphab
alphab=reshape(alphabito,length(alphabito),1);
%kalw th Huffman kai arxikopoiw to pinaka
pinakas=Huffman.empty(length(alphabito),0);
%dhmioyrgia tou alphabitou kai tw n pithanotitwn
for k=1:length(pithanotites)
    pinakas(k).probability = pithanotites(k);
    pinakas(k).character = alphab(k);
end

%-----%

% dhmioyrgia toy tree me tosa epipeda oso to to alphabito - 1
for k=1:size(pinakas,2)-1

    [~,point] = sort([pinakas.probability],'descend');%prwto kanw
    sortarisma tis pithanotites kata descending
    pinakas=pinakas(point);%dhmioyrgia neoy shmeiou
    kombos = Huffman; %dhmioyrgw neo kombo
    kombos.probability = pinakas(size(pinakas,2)).probability +
    pinakas(size(pinakas,2)-1).probability;%prosthetw th nea pithanotita
    pinakas(size(pinakas,2)).code = '0';%epeita eisagw to dyadiko
    kw dik 0
    pinakas(size(pinakas,2)-1).code = '1'; %epeita eisagw to dyadiko
    kw dik 1
    kombos.leftNode = pinakas(size(pinakas,2)-1);
    kombos.rightNode = pinakas(size(pinakas,2));
    pinakas = pinakas(1:size(pinakas,2)-2);%delete toys teleytaious 2
    pinakas = [kombos pinakas]; %o neos pinakas

end

rootNode = pinakas(1);%h riza tha einai to prwto stoixeio
kw d = char.empty;
```

```

eksodos=cell.empty;

%-----
%-----%
%kathe fora h dhmioyrgia toy tree ginetai epanlhptika apo
arxikopoiithe h
%riza kalw th synarthsh 'epanalipsi' dinontas ths ws eisodo th riza
to
%kwd kai thn eksodo

eksodos=epanalipsi(rootNode,kwd,eksodos);
kwdik=cell2table(eksodos,'VariableNames',{'Letter','Encoding','Propab
ility'});
kwdik=sortrows(kwdik,1);%sortarw tis grammes toy pinaka

end

function eksodos = epanalipsi (tempNode,kwd,eksodos)%ta pairnw ws
eisodo apo th huffmandict

%thelw na elegxw an exei paidia o trexwn kombos gia na stamathsei h
%anadromh. an den einai adeios-> exei paidia -> stamataw.
if ~isempty(tempNode)
kwd = [kwd tempNode.code]; % prosthetw to kwdiko pou me odhgei sta
fylla
kwd=strtrim(kwd);

%an o kombos einai xaraktiras tote sto f gyrnaei h epithymiti
kwdikopoihsh
if ~isempty(tempNode.character)
f = {tempNode.character , kwd , tempNode.probability};
eksodos=[ eksodos ; f];

end

%h epanalispi efarmozetai sto dexi k sto aristero kombo
eksodos=epanalipsi(tempNode.leftNode,kwd,eksodos);
eksodos=epanalipsi(tempNode.rightNode,kwd,eksodos);

end

end

και οι αρχικοποιήσεις Huffman tree classdef Huffman

%arxikopoihseis gia to dentro
properties

    leftNode = []
    rightNode = []
    probability
    code = ''
    character

end
    methods
end
end
end

```

β)

*Δημιουργία της συνάρτησης **huffmanencodex**:
κωδικοποιεί την ακολουθία απο σύμβολα(leksis) σε
δυναδικά ψηφία(leksien).*

```
function [ leksien ] = huffmanencodex( leksi , alphabito ,  
pithanotites )  
  
%dhmiourgia eisodou me th huffmandict  
eisodos = huffmandict(alphabito,pithanotites);  
%arxikopoihsh ths kwdikopoihmenhs leksis  
leksien='';  
  
%-----%  
  
%an dialekw na kwdikopoihsw ana dyadiko psifio  
if iscell(alphabito) == 0  
  
for p=1:length(leksi)  
  
    for k=1:length(alphabito)  
  
        if strcmp(leksi(p),eisodos.Letter(k))  
            %kwdikopoihsh  
            leksien=strcat(leksien,eisodos.Encoding(k));  
  
        end  
  
    end  
  
end  
  
%-----%  
  
%an dialekw na kwdikopoihsw me 2hs takshs epekstash  
else  
  
    for p=1:2:length(leksi) % bhma 2 -> 1:2:length  
  
        for k=1:length(alphabito)  
  
            %kwdikopoiw ana dyo -> i+1  
            if strcmp(leksi(p:p+1),eisodos.Letter(k))  
  
                leksien=strcat(leksien,eisodos.Encoding(k));  
  
            end  
  
        end  
  
    end  
  
end  
  
end
```

γ)

*Δημιουργία της συνάρτησης **huffmandecodex**:
αποκωδικοποιεί την ακολουθία από δυαδικά ψηφία(leksl) σε σύμβολα(lekside).*

```
function [ lekside ] = huffmandecodex( leksi , alphabito ,  
pithanotites )  
%prokeimenou na tairiazoun oi diastaseis kanw reshape sto yparxon kai  
to  
%anathetw sto alphab  
alphab=reshape(alphabito,length(alphabito),1);  
%kalw th Huffman kai arxikopoiw to pinaka  
pinakas=Huffman.empty(length(alphabito),0);  
%dhmioyrgia tou alphabitou kai tw n pithanotitwn  
for i=1:length(pithanotites)  
    pinakas(i).probability = pithanotites(i);  
    pinakas(i).character = alphab(i);  
end  
  
%-----  
-----%  
  
% dhmioyrgia toy tree me tosa epipeda oso to to alphabito - 1  
for e=1:size(pinakas,2)-1  
  
    [~,point] = sort([pinakas.probability],'descend');%prwto kanw  
sortarisma tis pithanotites kata descending  
    pinakas=pinakas(point);%dhmioyrgia neoy shmeiou  
    kombos = Huffman; %dhmioyrgw neo kombo  
    kombos.probability = pinakas(size(pinakas,2)).probability +  
pinakas(size(pinakas,2)-1).probability;%prosthetw th nea pithanotita  
    pinakas(size(pinakas,2)).code = '0';%epeita eisagw to dyadiko  
kwdiko 0  
    pinakas(size(pinakas,2)-1).code = '1'; %epeita eisagw to dyadiko  
kwdiko 1  
    kombos.leftNode = pinakas(size(pinakas,2)-1); %anathesi aristeros  
    kombos.rightNode = pinakas(size(pinakas,2)); %anathesi deksis  
    kombos  
    pinakas = pinakas(1:size(pinakas,2)-2);%delete toys teleytaious 2  
    pinakas = [kombos pinakas]; %o neos kombos sto pinaka  
  
end  
  
rootNode = pinakas(1);  
%me th char th metatrepw se xarakthres  
leksien = char(leksi);  
eksodos='';
```

```

lekside='';
%ta e dyadika pou xrhsimopoiithikan
[decleksi,e]=depanalipsi(leksien,rootNode,eksodos,0);

%-----
%-----%

while ~isempty(leksien)%otan einai keni ta exw perasei ola ta psifia
, oso den einai ta pernew
    leksien(1:e)=[];%diagrafi
    lekside=strcat(lekside,decleksi);%union of outputs
    [decleksi,e]=depanalipsi(leksien,rootNode,eksodos,0);%andromiki
klisi

end

end

end

end

end

%ΣΥΝΑΡΤΗΣΗ ΠΟΥ ΥΛΟΠΟΙΕΙ ΤΗΝ ΑΝΑΔΡΟΜΗ ΣΤΗΝ ΑΠΟΚΩΔΙΚΟΠΟΙΗΣΗ
function [ eksodos,e ] = depanalipsi( leksi ,tempNode,eksodos,e )

%trexon kombos->leksi pros kwdikopoihsh
if ~isempty(leksi) && ~isempty(tempNode.character)
    eksodos=tempNode.character;
    e=e;
    return

elseif isempty(leksi)
    eksodos=tempNode.character;
    e=e;
    return
end

%deksis kombos
%an to dyadiko psifio einai 0
%ayksanw ta psifia pou exoun prospelastei
%gia to gramma pou diabasa meiw nw thn eisodo
%kai deixnw sto deksi kombo
if ~isempty(tempNode) && strcmp(leksi(1),'0') &&
~isempty(tempNode.rightNode)
    e=e+1;
    leksi(1)=[];
    [eksodos,e]=depanalipsi(leksi,tempNode.rightNode,eksodos,e);

```

```

%aristeros kombos
%an to dyadiko psifio einai 1
%ayksanw ta psifia pou exoun prospelastei
%gia to gramma pou diabasa meiwnw thn eisodo
%kai deixnw ston aristero kombo
elseif ~isempty(tempNode) && strcmp(leksi(1),'1') &&
~isempty(tempNode.leftNode)
    e=e+1;
    leksi(1)=[];
    [eksodos,e]=depanalipsi(leksi,tempNode.leftNode,eksodos,e);

end

end

```


Ζητούμενο 2

*Αρχικά δημιουργούμε τη πρώτη πηγή (10000
χαρακτήρες)απο το αγγλικό αλφάβητο και τις
πιθανότητες απο τη Wikipedia όπως ακριβώς μας
υποδεικνύει η εκφώνηση της άσκησης .*

```
%%-----!!!!!!! Erwtima 2      !!!!!!!-----%%
clear
clc

%%%%----- 1h phgh-----%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%arxikopoihses gia sygkrisi
alphaeisa='';
dyadikoisa='';
elegxeisa='';

%pithanotites apo wikipedia
pithanotites=[0.08167  0.01492  0.02782  0.04253  0.12702  0.02228
0.02015 0.06094 0.06966  0.00153 0.00772 0.04025 0.02406 0.06749
0.07507 0.01929 0.00095 0.05987 0.06327 0.09056 0.02758 0.00978
0.02361 0.00150 0.01974 0.00074];
% arxikopoihsh to aggliko alfabito
alphab='abcdefghijklmnopqrstuvwxyz';
%26 theseis gia to kathe gramma
alphabin=[1:26];

%dhmiourgia random pigis A symfwna me ta dedomena
pigiA=randsrc(100,100,[alphabin;pithanotites]);
pigiA=reshape(pigiA,10000,1);
pigiA=num2cell(pigiA,[1 10000]);
%anathesi se cell array
Akelli = cell([10000 3]);

t = 1 ;
while t < size(pigiA{1,1},1)

    Akelli{t,1}=alphab(pigiA{1,1}(t));
    t = t+1 ;
end

k = 1 ;
while k < size(Akelli,1)

    alphaeisa=strcat(alphaeisa,Akelli{k,1});
    k = k+1;
end
```

```

%klisi ths huffmanencodex huffmandecodex me ta parapanw inputs
dyadikoisa=huffmanencodex(alphaeisa,alphab,pithanotites);
elegxeisa=huffmandecodex(dyadikoisa,alphab,pithanotites);

if strcmp(alphaeisa,elegxeisa)==1 %sygkrisi tw'n string an ==1 tote h
pigi A kwdikopoiei kai apodikwpoiei

    disp('Mpike sto if ara h pigi A einai ok!')

end

```

Δημιουργούμε τη δεύτερη πηγή από το withoutkeywords.txt το οποίο είναι το keywords.txt χωρίς ειδικούς χαρακτήρες και μόνο πεζά γράμματα τα οποία αφαιρέσαμε(με τη χρήση Notepad++) όπως υποδείχθηκε στο forum . Σαν αλφάβητο στη πηγή B έχουμε το ίδιο με της πηγής A.

```

%%%%----- 2h phgh----- exei to idio alphabito me thn 1h-----
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%arxikopoihseis kenwn char
alphaisb='';
dyadikoisb='';
elegxeisb='';

%anoigma me ton file pointer toy arxeiou pou dinetai
fid=fopen('withoutkeywords.txt','r');
Bkelli=textscan(fid,'%s');%diabasma
fclose(fid);%kleisimo

h = 1;
while h < size(Bkelli{1,1},1)

    alphaisb=strcat(alphaisb,Bkelli{1,1}{h,1});
    h = h+1;
end

%klisi ths huffmanencodex huffmandecodex me ta parapanw inputs
%gia th phgh b twra
dyadikoisb=huffmanencodex(alphaisb,alphab,pithanotites);
elegxeisb=huffmandecodex(dyadikoisb,alphab,pithanotites);

if strcmp(alphaisb,elegxeisb)==1 %sygkrisi tw'n string an ==1 tote h
pigi b kwdikopoiei kai apodikwpoiei
    disp(' Mpike sto if ara H pigi B einai ok!')
end

```

*Αφού κωδικοποιήσαμε τη πηγή A και B επιβεβαιώνουμε
κάθε φορά τη σωστή λειτουργία .*

```
Mpike sto if ara h pigi A einai ok!  
Mpike sto if ara H pigi B einai ok!
```

*Και στη συνέχεια θα πρέπει να εξετάσω το μήκος της
κωδικοποίησης :*

```
%to apotelesna ths huffmandict  
dictres=huffmandict(alphab,pithanotites);  
L=0;  
y = 1 ;  
while y < size(dictres,1)  
  
    chen=char(dictres.Encoding(y));  
    %ypologismos mikos l symfwna me ton typo  
    L=L+length(chen)*pithanotites(y);  
    y = y+1;  
end
```

```
>> L  
  
L =  
  
    4.1984
```

Όπως ήταν λογικό αφού και για τις δυο πηγές χρησιμοποιείται κοινό αλφάβητο και πιθανότητες το μέσο μήκος κωδικοποίησης είναι ίδιο . Το μέσο μήκος κωδικοποίησης δίνεται απο το τύπο :

$$L = \sum p(x)l(x)$$

Ζητούμενο 3

Στο ερώτημα αυτό δημιουργήσαμε τη πηγή B απο το *keywords.txt* που δόθηκε στην εκφώνηση χωρίς αλλαγές , ακριβώς όπως δίνεται .

```
%-----!!!!!!! Erwtima 3  !!!!!!!-----%%
clear
clc

    alphaeisa='';
    dyadikoisa='';
    elegxeisa='';

%-----pigi B -----
%anoigma arxeiou
fid=fopen('keywords.txt');
%diabasma
Bkelli=textscan(fid,'%s');
%kleisimo
fclose(fid);

% arxikopoihsh to aggliko alfabito mazi me ta kefalaia kai ta
symbola
alphab='-./'abcdefghijklmnopqrstuvwxyz';
```

Επειδή όμως δεν έχουμε τις πιθανότητες εμφάνισης όπως τονίστηκε στο forum απο τον υπεύθυνο θα πρέπει να υπολογίζουμε την πιθανότητα εμφάνισης κάθε συμβόλου .

```

% arxikopoihsh gia tis emfaniseis
events=zeros(length(alphab),1);
% arxikopoihsh gia tis pithanotites emnfanishs
pithanotites=zeros(length(alphab),1);
grammata=0;

for i = 1: length(alphab)

    for j = 1: size(Bkelli{1,1},1)
        %diatrexw gia na brw tis emfaniseis

events(i,1)=events(i,1)+sum(length(strfind(Bkelli{1,1}{j,1},alphab(i)
)));

    end

end

for x = 1: size(Bkelli{1,1},1)
    %ta grammata mesa sto keimeno
    grammata=grammata+length(Bkelli{1,1}{x,1});

end

for p = 1: length(alphab)
    %pithanotita emfanisis = arithmos / grammata
    pithanotites(p,1)=events(p,1)/grammata;

end

```

*στη συνέχεια όπως και στο B καλώ τις συναρτήσεις που
έφτιαξα στο ερώτημα 1*

```

for x = 1 : size(Bkelli{1,1},1)

    alphaeisa=strcat(alphaeisa,Bkelli{1,1}{x,1});

end

%klisi twn huffmanencodex kai huffmandecodex
dyadikoisa=huffmanencodex(alphaeisa,alphab,pithanotites);
elegxeisa=huffmandecodex(dyadikoisa,alphab,pithanotites);
%apotelesma ths huffmandict

```

```
dictres=huffmandict(alphab,pithanotites);  
dictres=sortrows(dictres,{'Propability'},{'ascend'});  
pithanotites=sort(pithanotites);  
L=0;
```

Και τέλος υπολογίζω το μέσο μήκος κώδικα. Βλέπουμε ότι σε σχέση με το προηγούμενο ερώτημα μειώθηκε το μέσο μήκος καθώς τα σύμβολα αυξήθηκαν. Η κατανομή των πιθανοτήτων για τον Huffman ήταν πιο κοντά στη πραγματικότητα όσον αφορά τη πηγή B.

```
for y = 1: size(dictres,1)  
  
    chen=char(dictres.Encoding(y));  
    %T meso mikos symfwna me ton typo  
    L=L+length(chen)*pithanotites(y)  
  
end
```

```
>> L  
  
L =  
  
4.1231
```

Ζητούμενο 4

Για 5000 χαρακτήρες χρησιμοποιούμε n -οστή επέκταση πηγής και τους παίρνουμε σε ζεύγη. Γι αυτούς υπολογίζουμε τις πιθανότητες εμφάνισης. Για n -οστή επέκταση πηγής και ακολουθία που έχουμε σε αυτό το ερώτημα θα έχουμε $L_n = L_n/2$

```
%-----%
%-----%
%%-----!!!!!!! Erwtima 4  !!!!!!!-----%%
clear
clc
%arxikopoihseis
alphaeisa='';
dyadikoisa='';
elegxeisa='';

%pithanotites apo wikipedia
pithanotites=[0.08167 0.01492 0.02782 0.04253 0.12702 0.02228
0.02015 0.06094 0.06966 0.00153 0.00772 0.04025 0.02406 0.06749
0.07507 0.01929 0.00095 0.05987 0.06327 0.09056 0.02758 0.00978
0.02361 0.00150 0.01974 0.00074];
% arxikopoihsh to aggliko alfabito
alfavito='abcdefghijklmnopqrstuvwxyz';

%2hs taksis epektasi pigis
alphab=cell.empty(length(alfavito)^2,0);
pithanotites=zeros(length(alfavito)^2,1);
a=0;

for i=1:length(alfavito)

    for j=1:length(alfavito)

        alphab{a+j,1}=[alfavito(i) alfavito(j)];
```



```

pithanotites(a+j,1)=pithanothtes(i)*pithanothtes(j);

end
%sympilorwnw kata a ta kelia wste na parw ena dyo toys xaraktires
a=a+j;

end

alphain=1:1:size(alphab,1);
%pigi A me 5000 zeygi
pigiA=randsrc(5000,1,[alphain;pithanotites']);
Akelli=cell([5000 3]);

for i=1:length(pigiA)

    Akelli{i,1} = alphab(pigiA(i,1));

end

for a=1:size(Akelli,1)

    alphaeisa=strcat(alphaeisa,Akelli{a,1});

end
%casting se char
alphaeisa=char(alphaeisa);

dyadikoisa=huffmanencodex(alphaeisa,alphab,pithanotites);
elegxeisa=huffmandecodex(dyadikoisa,alphab,pithanotites);

%apotelesma ths huffmandictation
dictres=huffmandict(alphab,pithanotites);
L=0;
s=1;
while s < size(dictres,1)

    chen=char(dictres.Encoding(s));
    %typos MESO MHKOS
    L=L+length(chen)*pithanotites(s)

    s = s+1 ;

end

%L = Ln /2  n-osti epektasi
L=L/2

```

παρατηρούμε πως το μήκος κωδικοποίησης σε σχέση με
ζητούμενο 2 μειώθηκε λόγω του ότι πήραμε τους
χαρακτήρες ως ζεύγη, ωστόσο όχι αισθητά (ήταν 4.1984)

```
>> L
L =
    4.1910
fx >>
```

Ζητούμενο 5

*Επειδή χρησιμοποιήσαμε τις πιθανότητες του
προηγούμενου ερωτήματος περιμένουμε να έχουμε τα
ίδια αποτελέσματα*

```
%%-----!!!!!!! Erwtima 5  !!!!!!!-----%%
clear
clc
%arxikopoihseis
alphaeisa='';
dyadikoisa='';
elegxeisa='';

%pithanotites apo wikipedia
pithanotites=[0.08167  0.01492  0.02782  0.04253  0.12702  0.02228
0.02015 0.06094 0.06966  0.00153 0.00772 0.04025 0.02406 0.06749
0.07507 0.01929 0.00095 0.05987 0.06327 0.09056 0.02758 0.00978
0.02361 0.00150 0.01974 0.00074];
% arxikopoihsh to aggliko alfabito
alfavito='abcdefghijklmnopqrstuvwxyz';

%2hs taksis epektasi pigis
alphab=cell.empty(length(alfavito)^2,0);
pithanotites=zeros(length(alfavito)^2,1);
a=0;

for i=1:length(alfavito)

    for j=1:length(alfavito)

        alphab{a+j,1}=[alfavito(i) alfavito(j)];
        pithanotites(a+j,1)=pithanotites(i)*pithanotites(j);

    end
    %sympyorwnw kata a ta kelia wste na parw ena dyo toys xaraktires
    a=a+j;

end
%anoigma arxeiou
fp=fopen('keywords.txt');
%diabasma
Bkelli=textscan(fp,'%s');
%kleisimo
fclose(fp);
```

```

a = 0 ;
for a=1:size(Bkelli,1)

    alphaeisa=strcat(alphaeisa,Bkelli{a,1});

end
%casting se char
alphaeisa=char(alphaeisa);

dyadikoisa=huffmanencodex(alphaeisa,alphab,pithanotites);
elegxeisa=huffmandecodex(dyadikoisa,alphab,pithanotites);

%apotelesma ths huffmandictation
dictres=huffmandict(alphab,pithanotites);
L=0;
s=1;
while s < size(dictres,1)

    chen=char(dictres.Encoding(s));
    %typos MESO MHKOS
    L=L+length(chen)*pithanotites(s)

    s = s+1 ;
end

%L = Ln /2  n-osti epektasi
L=L/2;

```

όπως αναμενόταν έχουμε και εδώ τις ίδιες μετρήσεις.

```

>> L

L =

    4.1910

>> |

```

*Αν τώρα χρησιμοποιήσω ως πιθανότητες τις εμφανίσεις
των χαρακτήρων του αλφαβήτου μέσα στο κείμενο*

```
%anoigma arxeiou me ola ta symbola to arxiko
fid=fopen('keywords.txt');
kelliB=textscan(fid,'%s');
fclose(fid);

%arxikopoihseis
alphaeisa='';
dyadikoisa='';
elegxeisa='';

%arxikopoihsh alfabitou me ola ta symbola
alphabito='-./'abcdefghijklmnopqrstuvwxyz';

alphaB=cell.empty(length(alphabito)^2,0);
pithanotites=zeros(length(alphabito)^2,1);
% arxikopoihsh gia tis emfaniseis
emfaniseis=zeros(length(alphabito),1);
% arxikopoihsh gia tis pithanotites emnfanishs
pith=zeros(length(alphabito),1);
grammata=0;

for n=1:length(alphabito)

    for m=1:size(kelliB{1,1},1)
        %diatrexw gia na brw tis emfaniseis

emfaniseis(n,1)=emfaniseis(n,1)+sum(length(strfind(kelliB{1,1}{m,1},a
lphabito(n))));

    end

end

for a=1:size(kelliB{1,1},1)
    %ta grammata mesa sto keimeno
```

```

        grammata=grammata+length(kelliB{1,1}{a,1});
end

for a=1:length(alphabito)

    pith(a,1)=emfaniseis(a,1)/grammata;

end

a=0;

for n=1:length(alphabito)

    for m=1:length(alphabito)

        alphaB{a+m,1}=[alphabito(n) alphabito(m)];
        pithanotites(a+m,1)=pith(n)*pith(m);

    end

    a=a+m;%arithmos gia sumplhrwsh keliwn

end

for a=1:size(kelliB{1,1},1)
    %pithanotita emfanisis = arithmos / grammata

    alphaeisa=strcat(alphaeisa,kelliB{1,1}{a,1});

end

%klisi tw'n huffmanencodex kai huffmandecodex

dyadikoisa=huffmanencodex(alphaeisa,alphaB,pithanotites);
dyadikoisa=huffmandecodex(dyadikoisa,alphaB,pithanotites);

%apotelesma ths huffmandict

dictres=huffmandict(alphaB,pithanotites);
dictres=sortrows(dictres,{'Propability'},{'ascend'});
pithanotites=sort(pithanotites);
L=0;

for a=1:size(dictres,1)
    %T meso mikos symfwna me ton typo

    chen=char(dictres.Encoding(a));
    L=L+length(chen)*pithanotites(a);

end

L=L/2;

```

Σύμφωνα με το αποτέλεσμα στη matlab το μήκος έχει μειωθεί σε σχέση με πριν.

```
>> L  
  
L =  
  
    4.0966  
fx >> |
```

Τελικό συμπέρασμα

Σκοπός της άσκησης ήταν να δούμε το μήκος κωδικοποίησης σε κάθε ερώτημα και συγκρίσεις μεταξύ τους. Θέλουμε να δούμε αν τελικά συμφέρει να πάρεις ένα γενικό αλφάβητο ή ένα αλφάβητο που προκύπτει ανάλογα με την πηγή. Βλέπουμε ότι συμφέρει τελικά να ασχολούμαστε με το είδος της πηγής και να κωδικοποιούμε την πηγή με ζεύγη συμβόλων αν και εδώ δεν ήταν όσο αποδοτικό θα περιμέναμε. Σε κάθε περίπτωση εξετάσαμε τί δίνει καλύτερα αποτελέσματα.