

## Laboratorijska vježba 02 : Shannon-Fano algoritam kodiranja

---

Za izradu laboratorijske vježbe treba koristiti odgovarajuću Jupyter Notebook datoteku. Urađenu vježbu je potrebno konvertirati u PDF format, a zatim je PDF datoteku potrebno predati do postavljenog roka koristeći platformu Zamger.

Ime i prezime studenta, broj indeksa:

Amar Hasečić, 2116/18673

Datum izrade izvještaja:

19.3.2024.

---

### Zadatak 1.

Potrebno je realizirati Shannon-Fano algoritam kodiranja putem funkcije `shannon_fano` koja kao parametar prima neku tekstualnu poruku, a kao rezultat vraća listu simbola i listu svih kodova dodijeljenih pojedinačnim simbolima. Potrebno je implementirati i funkcije `kodiraj` i `entropija`. Funkcija `kodiraj` kao parametre prima poruku, listu simbola i listu kodova, a kao rezultat vraća kodiranu poruku. Funkcija `entropija` kao parametar prima poruku, a kao rezultat vraća entropiju za tu poruku.

Shannon-Fano algoritam predstavlja osnovni algoritam kodiranja na bazi vjerovatnoće pojavljivanja simbola u poruci. Sastoji se iz sljedećih koraka:

1. Za zadanu listu simbola, izračunati vjerovatnoće pojavljivanja u poruci.
2. Sortirati simbole prema frekvenciji od najveće do najmanje frekvencije.
3. Podijeliti listu na dva dijela, po mogućnosti sa jednakim sumama vjerovatnoća ponavljanja.
4. Lijeva ivica dobiva vrijednost 0, a desna 1 (svi kodovi koji počinju sa simbolima na lijevoj strani počinju sa 0, a na desnoj sa 1).
5. Izvršavati korake 3 i 4 dok se potpuno ne izgradi binarno stablo (svi simboli postanu čvorovi stabla).

Detaljne informacije o samom algoritmu moguće je pronaći u Poglavlju 2 materijala za rad na predmetu, na str. 21 - 25.

Dobivena rješenja za kodne riječi nisu jedinstvena, tj. dobivena rješenja za kodne riječi ovise o specifičnim implementacijskim detaljima. Bitno je da implementacija bude realizirana u skladu sa gore opisanim koracima, tj. u skladu sa algoritmom opisanim na predavanjima.

### Rješenje:

```
def dodaj_kodove(simbol_p_kod, start, end, prefix=''):
    if start == end:
        return
    elif start == end - 1:
        if len(simbol_p_kod) == 1:
            simbol_p_kod[start] = (simbol_p_kod[start][0],
simbol_p_kod[start][1], '0')
        else:
            simbol_p_kod[start] = (simbol_p_kod[start][0],
simbol_p_kod[start][1], prefix)
    else:
        ukupna_vjerovatnoca = sum(simbol[1] for simbol in
simbol_p_kod[start:end])
        kumulativna_vjerovatnoca = 0
        split_index = start
        for i in range(start, end):
            kumulativna_vjerovatnoca += simbol_p_kod[i][1]
            if kumulativna_vjerovatnoca >= ukupna_vjerovatnoca /
2:
                split_index = i
                break
        for i in range(start, split_index + 1):
            simbol_p_kod[i] = (simbol_p_kod[i][0], simbol_p_kod[i]
[1], prefix + '0' + simbol_p_kod[i][2])
        for i in range(split_index + 1, end):
            simbol_p_kod[i] = (simbol_p_kod[i][0], simbol_p_kod[i]
[1], prefix + '1' + simbol_p_kod[i][2])
        dodaj_kodove(simbol_p_kod, start, split_index + 1, prefix
+ '0')
        dodaj_kodove(simbol_p_kod, split_index + 1, end, prefix +
'1')

def shannon_fano(poruka):
    simboli = list(set(poruka))
    vjerovatnoce = [0]*len(simboli)
    kodovi = ['']*len(simboli)

    for i in range(len(simboli)):
        for simbol in poruka:
```

```

        if simboli[i] == simbol:
            vjerovatnoce[i] += 1
        vjerovatnoce[i] /= len(poruka)

    simbol_p_kod = []
    for i in range(len(simboli)):
        simbol_p_kod.append((simboli[i], vjerovatnoce[i], kodovi[i]))

    simbol_p_kod = sorted(simbol_p_kod, key=lambda x: x[0]) # po
alfabetu
    simbol_p_kod = sorted(simbol_p_kod, key=lambda x: x[1],
reverse=True)

    dodaj_kodove(simbol_p_kod, 0, len(simbol_p_kod))

    return [item[0] for item in simbol_p_kod], [item[2] for item in
simbol_p_kod]

def kodiraj(poruka, simboli, kodovi):
    kodirana = ""
    for simbol in poruka:
        kodirana += kodovi[simboli.index(simbol)]
    return kodirana

import math
def entropija(poruka):

    simboli = list(set(poruka))
    vjerovatnoce = [0] * len(simboli)
    for i in range(len(simboli)):
        for simbol in poruka:
            if simboli[i] == simbol:
                vjerovatnoce[i] += 1
        vjerovatnoce[i] /= len(poruka)

    rezultat = 0

    for v in vjerovatnoce:
        rezultat += v * math.log(v, 2)

    rezultat *= -1

    return abs(rezultat)

```

Nakon implementacije funkcije, potrebno je biti moguće izvršiti programski kod ispod tako da daje prikazani ispis. Osim toga, potrebno je pored poruke "CAADADBAADCAADBAADAD" dodati još četiri primjera poruka kako bi se implementacija testirala. Dodatna četiri primjera poruka

trebaju obavezno biti iz sljedećeg skupa {"A", "DDDDDDDDDDDDDDDDDDDD", "ABBBBCDECCCAAACAAAAAAAAAAAA", "AAAAAAAAABBBBBBCCCD"}.

```
import math
poruka = "CAADADBAADCAADBAADAD"
print("Izvorna poruka je:")
print(poruka)

(simboli, kodovi) = shannon_fano(poruka)

print("\nPoruka se sastoji od sljedećih simbola:")
print(", ".join(map(str, simboli)))

print("\nSkup dobivenih kodnih riječi je:")
print(", ".join(map(str, kodovi)))

kodirana = kodiraj(poruka, simboli, kodovi)

for i in range(len(simboli)):
    print("\nKod simbola " + simboli[i] + " je " + kodovi[i])

print("\nKodirana poruka je:")
print(kodirana)
print("\nEntropija poruke je:")
print(round(entropija(poruka),3))
print("\nProsječna dužina kodne riječi u poruci je:")
print(round(len(kodirana)/len(poruka),3))
```

Izvorna poruka je:  
CAADADBAADCAADBAADAD

Poruka se sastoji od sljedećih simbola:  
A, D, B, C

Skup dobivenih kodnih riječi je:  
0, 10, 110, 111

Kod simbola A je 0

Kod simbola D je 10

Kod simbola B je 110

Kod simbola C je 111

Kodirana poruka je:  
11100100101110001011100101100010010

Entropija poruke je:  
1.685

Prosječna dužina kodne riječi u poruci je:  
1.7

*#Test 1*

```
import math
```

```
poruka = "A"
```

```
print("Izvorna poruka je:")
```

```
print(poruka)
```

```
(simboli, kodovi) = shannon_fano(poruka)
```

```
print("\nPoruka se sastoji od sljedećih simbola:")
```

```
print(", ".join(map(str, simboli)))
```

```
print("\nSkup dobivenih kodnih riječi je:")
```

```
print(", ".join(map(str, kodovi)))
```

```
kodirana = kodiraj(poruka, simboli, kodovi)
```

```
for i in range(len(simboli)):
```

```
    print("\nKod simbola " + simboli[i] + " je " + kodovi[i])
```

```
print("\nKodirana poruka je:")
```

```
print(kodirana)
```

```
print("\nEntropija poruke je:")
```

```
print(round(entropija(poruka),3))
```

```
print("\nProsječna dužina kodne riječi u poruci je:")
```

```
print(round(len(kodirana)/len(poruka),3))
```

Izvorna poruka je:

A

Poruka se sastoji od sljedećih simbola:

A

Skup dobivenih kodnih riječi je:

0

Kod simbola A je 0

Kodirana poruka je:

0

Entropija poruke je:

0.0

Prosječna dužina kodne riječi u poruci je:

1.0

*#Test 2*

```

import math
poruka = "DDDDDDDDDDDDDDDDDDDD"
print("Izvorna poruka je:")
print(poruka)

(simboli, kodovi) = shannon_fano(poruka)

print("\nPoruka se sastoji od sljedećih simbola:")
print(", ".join(map(str, simboli)))

print("\nSkup dobivenih kodnih riječi je:")
print(", ".join(map(str, kodovi)))

kodirana = kodiraj(poruka, simboli, kodovi)

for i in range(len(simboli)):
    print("\nKod simbola " + simboli[i] + " je " + kodovi[i])

print("\nKodirana poruka je:")
print(kodirana)
print("\nEntropija poruke je:")
print(round(entropija(poruka),3))
print("\nProsječna dužina kodne riječi u poruci je:")
print(round(len(kodirana)/len(poruka),3))

```

Izvorna poruka je:  
DDDDDDDDDDDDDDDDDDDD

Poruka se sastoji od sljedećih simbola:  
D

Skup dobivenih kodnih riječi je:  
0

Kod simbola D je 0

Kodirana poruka je:  
000000000000000000

Entropija poruke je:  
0.0

Prosječna dužina kodne riječi u poruci je:  
1.0

### #Test 3

```

import math
poruka = "ABBBBCDECCCAAAAAAAAAAAAAAAAAA"
print("Izvorna poruka je:")
print(poruka)

```

```

(simboli, kodovi) = shannon_fano(poruka)

print("\nPoruka se sastoji od sljedećih simbola:")
print(", ".join(map(str, simboli)))

print("\nSkup dobivenih kodnih riječi je:")
print(", ".join(map(str, kodovi)))

kudirana = kodiraj(poruka, simboli, kodovi)

for i in range(len(simboli)):
    print("\nKod simbola " + simboli[i] + " je " + kodovi[i])

print("\nKudirana poruka je:")
print(kudirana)
print("\nEntropija poruke je:")
print(round(entropija(poruka), 3))
print("\nProsječna dužina kodne riječi u poruci je:")
print(round(len(kudirana)/len(poruka), 3))

```

Izvorna poruka je:  
 ABBBBBCDECCCCAAAAAAAAAAAAAAAAA

Poruka se sastoji od sljedećih simbola:  
 A, C, B, D, E

Skup dobivenih kodnih riječi je:  
 0, 100, 101, 110, 111

Kod simbola A je 0

Kod simbola C je 100

Kod simbola B je 101

Kod simbola D je 110

Kod simbola E je 111

Kudirana poruka je:  
 0101101101101100110111100100100100000000000000000

Entropija poruke je:  
 1.692

Prosječna dužina kodne riječi u poruci je:  
 1.846

#Test 4

```

import math
poruka = "AAAAAAAAAABBBBBBCCCD"
print("Izvorna poruka je:")
print(poruka)

(simboli, kodovi) = shannon_fano(poruka)

print("\nPoruka se sastoji od sljedećih simbola:")
print(", ".join(map(str, simboli)))

print("\nSkup dobivenih kodnih riječi je:")
print(", ".join(map(str, kodovi)))

kodirana = kodiraj(poruka, simboli, kodovi)

for i in range(len(simboli)):
    print("\nKod simbola " + simboli[i] + " je " + kodovi[i])

print("\nKodirana poruka je:")
print(kodirana)
print("\nEntropija poruke je:")
print(round(entropija(poruka),3))
print("\nProsječna dužina kodne riječi u poruci je:")
print(round(len(kodirana)/len(poruka),3))

```

Izvorna poruka je:  
 AAAAAAAAAAABBBBBBCCCD

Poruka se sastoji od sljedećih simbola:  
 A, B, C, D

Skup dobivenih kodnih riječi je:  
 0, 10, 110, 111

Kod simbola A je 0

Kod simbola B je 10

Kod simbola C je 110

Kod simbola D je 111

Kodirana poruka je:  
 0000000000101010101010110110110111

Entropija poruke je:  
 1.648

Prosječna dužina kodne riječi u poruci je:  
 1.7