

[6 bodova] Provedite 2 iteracije (2 izmjene baze gdje se inicijalna baza ne računa) simpleks algoritma nad dolje zadanim linearnim programom.

$$\min -4x_1 - 3x_2 + 1x_3$$

$$-4x_1 + 3x_2 + 3x_3 \leq 8$$

$$4x_1 + 4x_2 - 5x_3 \leq 7$$

$$-2x_1 + 4x_2 + x_3 \leq 7$$

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0$$

Kao rješenje nakon zadanog broja iteracija unesite:

- status rješenja problema kao varijablu **status** značenja: -1 za detektirani neizvediv problem, 0 za detektirani neograničen problem, 1 za optimalno riješen problem, 2 za još neutvrđeni status (treba daljnje iteracije)
- Vašu konačnu simpleks tablicu u dvodimenzionalno polje **tableau**. Polje je izraženo kao lista sastavljena od podlista u jeziku Python,
- vektor konačne točke rješenje kao listu **solution** u jeziku Python (treba sadržavati vrijednosti **svih** varijabli u simpleks tablici),
- iznos optimalne fje cilja polaznog problema kao varijablu **obj** koja sadrži decimalni broj

Ako je status -1 ili 0, neće se gledati evaluirati vrijednosti unutar **solution** i **obj**, ali ih svejedno definirajte u rješenju na prave dimenzije (bitno za evaluator) i popunite sa nulama. Ako je status 2, unesite trenutno rješenje i njegovu vrijednost fje cilja.

Ako dobijete konačno rješenje u manjem broju iteracija od zadanog, koristite to rješenje za unos.

Numerička tolerancija je 10^{-2} .

Primjer valjanog formata unosa rješenja (nije točno rješenje zadatka, čak niti veličine dimenzija):

```
status = 1

tableau = [
[0,0,1,2,20],
[1,0,-3,7,8.8],
[0,1,4,-2.3,23]
]

solution = [0,0,0,0]

obj = -1
```

Napomena

- U zadacima **ne smijete** dodavati dodatne varijable, funkcije i sl.
 - Očekuju se samo tražene varijable
 - Morate definirati sve varijable, makar prazne, u suprotnom će vam padati testovi!
 - Ako imate dodatnih varijabli, funkcija ili klasa, padat će vam svi testovi!
- Fotografije postupka ovog zadatka uploadate u odvojenom ispitu
 - Dobit ćete 0 bodova ako nemate uploadane fotografije postupka rješavanja ovog zadatka sa potpisanog papira

[6 bodova] Provedite 2 iteracije (2 izmjene baze gdje se inicijalna baza ne računa) simpleks algoritma nad dolje zadanim linearnim programom.

$$\max 4x_1 + 3x_2 - 1x_3$$

$$-4x_1 + 3x_2 + 3x_3 \leq 8$$

$$4x_1 + 4x_2 - 5x_3 \leq 7$$

$$-2x_1 + 4x_2 + x_3 \leq 7$$

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0$$

Kao rješenje nakon zadanog broja iteracija unesite:

- status rješenja problema kao varijablu **status** značenja: -1 za detektirani neizvediv problem, 0 za detektirani neograničen problem, 1 za optimalno riješen problem, 2 za još neutvrđeni status (treba daljnje iteracije)
- Vašu konačnu simpleks tablicu u dvodimenzionalno polje **tableau**. Polje je izraženo kao lista sastavljena od podlista u jeziku python,
- vektor konačne točke rješenje kao listu **solution** u jeziku python (treba sadržavati vrijednosti **svih** varijabli u simpleks tablici),
- iznos optimalne fje cilja polaznog problema kao varijablu **obj** koja sadrži decimalni broj

Ako je status -1 ili 0, neće se gledati evaluirati vrijednosti unutar **solution** i **obj**, ali ih svejedno definirajte u rješenju na prave dimenzije (bitno za evaluator) i popunite sa nulama. Ako je status 2, unesite trenutno rješenje i njegovu vrijednost fje cilja.

Ako dobijete konačno rješenje u manjem broju iteracija od zadanog, koristite to rješenje za unos.

Numerička tolerancija je 10^{-2} .

Primjer valjanog formata unosa rješenja (nije točno rješenje zadatka, čak niti veličine dimenzija):

```
status = 1

tableau = [
[0,0,1,2,20],
[1,0,-3,7,8.8],
[0,1,4,-2.3,23]
]

solution = [0,0,0,0]

obj = -1
```

Napomena

- U zadacima **ne smijete** dodavati dodatne varijable, funkcije i sl.
 - Očekuju se samo tražene varijable
 - Morate definirati sve varijable, makar prazne, u suprotnom će vam padati testovi!
 - Ako imate dodatnih varijabli, funkcija ili klasa, padat će vam svi testovi!
- Fotografije postupka ovog zadatka uploadate u odvojenom ispitu
 - Dobit ćete 0 bodova ako nemate uploadane fotografije postupka rješavanja ovog zadatka sa potpisanog papira

[6 bodova] Provedite 2 iteracije(2 izmjene baze gdje se inicijalna baza ne računa) simpleks algoritma nad dolje zadanim linearnim programom.

$$\min -x_1 - 4x_2 - 2x_3$$

$$4x_1 + x_2 - x_3 \leq 6$$

$$4x_1 - 3x_2 + 4x_3 \leq 1$$

$$-3x_1 - x_2 - 4x_3 \leq 7$$

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0$$

Kao rješenje nakon zadanog broja iteracija unesite:

- status rješenja problema kao varijablu **status** značenja: -1 za detektirani neizvediv problem, 0 za detektirani neograničen problem, 1 za optimalno riješen problem, 2 za još neutvrđeni status (treba daljnje iteracije)
- Vašu konačnu simpleks tablicu u dvodimenzionalno polje **tableau**. Polje je izraženo kao lista sastavljena od podlista u jeziku python,
- vektor konačne točke rješenje kao listu **solution** u jeziku python (treba sadržavati vrijednosti **svih** varijabli u simpleks tablici),
- iznos optimalne fje cilja polaznog problema kao varijablu **obj** koja sadrži decimalni broj

Ako je status -1 ili 0, neće se gledati evaluirati vrijednosti unutar **solution** i **obj**, ali ih svejedno definirajte u rješenju na prave dimenzije (bitno za evaluator) i popunite sa nulama. Ako je status 2, unesite trenutno rješenje i njegovu vrijednost fje cilja.

Ako dobijete konačno rješenje u manjem broju iteracija od zadanog, koristite to rješenje za unos.

Numerička tolerancija je 10^{-2} .

Primjer valjanog formata unosa rješenja (nije točno rješenje zadatka, čak niti veličine dimenzija):

```
status = 1

tableau = [
[0,0,1,2,20],
[1,0,-3,7,8.8],
[0,1,4,-2.3,23]
]

solution = [0,0,0,0]

obj = -1
```

Napomena

- U zadacima **ne smijete** dodavati dodatne varijable, funkcije i sl.
 - Očekuju se samo tražene varijable
 - Morate definirati sve varijable, makar prazne, u suprotnom će vam padati testovi!
 - Ako imate dodatnih varijabli, funkcija ili klasa, padat će vam svi testovi!
- Fotografije postupka ovog zadatka uploadate u odvojenom ispitu
 - Dobit ćete 0 bodova ako nemate uploadane fotografije postupka rješavanja ovog zadatka sa potpisanog papira

[B-stablo](6 bodova)

U inicijalno prazno B-stablo **stupnja 3** redom unesite sljedeće brojeve:

27, 62, 73, 18, 56, 16, 7, 54, 9, 8, 76, 22

Kako svaki čvor u B-stablu može sadržati više vrijednosti, u Edgaru ćete svaki čvor prikazati u obliku ugnježđenih Python lista (*liste lista*). **Poredak ugnježđenih lista nije bitan**, ali samih **elemenata unutar** ugnježđenih lista (čvorova) jest.

Pitanja

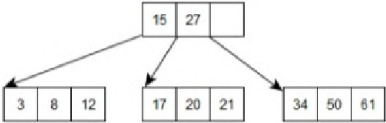
- a) Nakon unosa **16** prikažite **listove** - varijabla **odg1**
- b) Nakon unosa **76** prikažite **unutarnje čvorove** - varijabla **odg2**
- c) Nakon unosa **svih** prikažite **unutarnje čvorove** - varijabla **odg3**
- d) Nakon unosa **svih** prikažite **listove** - varijabla **odg4**

Primjer valjanog formata rješenja (nije točno rješenje zadatka)

```
odg1 = [[1, 2, 3], [4, 5, 6]]
odg2 = [[3, 4, 5], [6, 7, 8]]
odg3 = [[1, 2], [9, 10], [11]]
odg4 = [[3, 4], [5, 6], [7]]
```

Primjer

Za naredno B-stablo:



i ove željene odgovore

- **odg1** - Svi unutarnji čvorovi
- **odg2** - Svi listovi

dobivamo rješenje:

```
odg1 = [[15, 27]]
odg2 = [[3, 8, 12], [34, 50, 61], [17, 20, 21]]
# Poredak pojedinih čvorova smije biti zamijenjen, ali
# poredak unutar samih čvorova, mora biti valjan.
```

Napomena

- U zadatku se od vas traži da upisujete međurezultate. Pročitajte zadatak u cijelosti prije početka rješavanja jer bi se vaše stablo moglo bitno izmjeniti nakon dodavanja/brisanja dodatnih elemenata.
- U zadacima **ne smijete** dodavati vlastite (dodatne) varijable, funkcije i sl.
 - Očekuju se samo tražene varijable (*odg1, odg2, ...*)
 - Definirajte sve varijable, makar prazne, u suprotnom vam mogu testovi padati!
 - Ako imate dodatnih, vlastitih varijabli, funkcija ili klasa, padat će vam svi testovi!
- Fotografije postupka ovog zadatka uploadate u odvojenom ispitu
 - Dobit ćete 0 bodova ako nemate uploadane fotografije postupka rješavanja ovog zadatka sa potpisanog papira

[B-stablo](6 bodova)

U inicijalno prazno B-stablo **stupnja 4** redom unesite sljedeće brojeve:

74, 57, 34, 36, 46, 42, 8, 38, 15, 58, 23, 14, 52, 39

Kako svaki čvor u B-stablu može sadržati više vrijednosti, u Edgaru ćete svaki čvor prikazati u obliku ugnježđenih Python lista (*liste lista*). **Poredak ugnježđenih lista nije bitan**, ali samih **elemenata unutar** ugnježđenih lista (čvorova) jest.

Pitanja

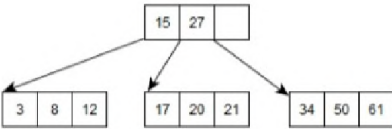
- a) Nakon unosa **38** prikažite **listove** - varijabla **odg1**
- b) Nakon unosa **52** prikažite **unutarnje čvorove** - varijabla **odg2**
- c) Nakon unosa **svih** prikažite **unutarnje čvorove** - varijabla **odg3**
- d) Nakon unosa **svih** prikažite **listove** - varijabla **odg4**

Primjer valjanog formata rješenja (nije točno rješenje zadatka)

```
odg1 = [[1, 2, 3], [4, 5, 6]]
odg2 = [[3, 4, 5], [6, 7, 8]]
odg3 = [[1, 2], [9, 10], [11]]
odg4 = [[3, 4], [5, 6], [7]]
```

Primjer

Za naredno B-stablo:



i ove željene odgovore

- **odg1** - Svi unutarnji čvorovi
- **odg2** - Svi listovi

dobivamo rješenje:

```
odg1 = [[15, 27]]
odg2 = [[3, 8, 12], [34, 50, 61], [17, 20, 21]]
# Poredak pojedinih čvorova smije biti zamijenjen, ali
# poredak unutar samih čvorova, mora biti valjan.
```

Napomena

- U zadatku se od vas traži da upisujete međurezultate. Pročitajte zadatak u cijelosti prije početka rješavanja jer bi se vaše stablo moglo bitno izmjeniti nakon dodavanja/brisanja dodatnih elemenata.
- U zadacima **ne smijete** dodavati vlastite (dodatne) varijable, funkcije i sl.
 - Očekuju se samo tražene varijable (*odg1, odg2, ...*)
 - Definirajte sve varijable, makar prazne, u suprotnom vam mogu testovi padati!
 - Ako imate dodatnih, vlastitih varijabli, funkcija ili klasa, padat će vam svi testovi!
- Fotografije postupka ovog zadatka uploadate u odvojenom ispitu
 - Dobit ćete 0 bodova ako nemate uploadane fotografije postupka rješavanja ovog zadatka sa potpisanog papira

[B-stablo](6 bodova)

U inicijalno prazno B-stablo **stupnja 3** redom unesite sljedeće brojeve:

20, 50, 36, 47, 21, 22, 45, 6, 61, 46, 79, 76

Kako svaki čvor u B-stablu može sadržati više vrijednosti, u Edgaru ćete svaki čvor prikazati u obliku ugnježđenih Python lista (*liste lista*). **Poredak ugnježđenih lista nije bitan**, ali samih **elemenata unutar** ugnježđenih lista (čvorova) jest.

Pitanja

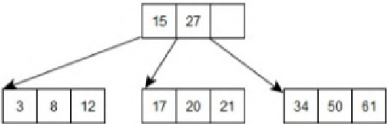
- a) Nakon unosa **45** prikažite **listove** - varijabla **odg1**
- b) Nakon unosa **79** prikažite **unutarnje čvorove** - varijabla **odg2**
- c) Nakon unosa **svih** prikažite **unutarnje čvorove** - varijabla **odg3**
- d) Nakon unosa **svih** prikažite **listove** - varijabla **odg4**

Primjer valjanog formata rješenja (nije točno rješenje zadatka)

```
odg1 = [[1, 2, 3], [4, 5, 6]]
odg2 = [[3, 4, 5], [6, 7, 8]]
odg3 = [[1, 2], [9, 10], [11]]
odg4 = [[3, 4], [5, 6], [7]]
```

Primjer

Za prikazano B-stablo:



i ove željene odgovore

- **odg1** - Svi unutarnji čvorovi
- **odg2** - Svi listovi

dobivamo rješenje:

```
odg1 = [[15, 27]]
odg2 = [[3, 8, 12], [34, 50, 61], [17, 20, 21]]
# Poredak pojedinih čvorova smije biti zamijenjen, ali
# poredak unutar samih čvorova, mora biti valjan.
```

Napomena

- U zadatku se od vas traži da upisujete međurezultate. Pročitajte zadatak u cijelosti prije početka rješavanja jer bi se vaše stablo moglo bitno izmjeniti nakon dodavanja/brisanja dodatnih elemenata.
- U zadacima **ne smijete** dodavati vlastite (dodatne) varijable, funkcije i sl.
 - Očekuju se samo tražene varijable (*odg1, odg2, ...*)
 - Definirajte sve varijable, makar prazne, u suprotnom vam mogu testovi padati!
 - Ako imate dodatnih, vlastitih varijabli, funkcija ili klasa, padat će vam svi testovi!
- Fotografije postupka ovog zadatka uploadate u odvojenom ispitu
 - Dobit ćete 0 bodova ako nemate uploadane fotografije postupka rješavanja ovog zadatka sa potpisanog papira

[AVL stablo](6 bodova)

U inicijalno prazno AVL stablo redom unesite sljedeće brojeve:

15, 12, 4, 41, 21, 39, 34, 17

Pitanja

- a) Nakon unosa **41** prikažite **listove** - varijabla **odg1**
- b) Nakon unosa **39** prikažite **unutarnje čvorove** - varijabla **odg2**
- c) Nakon unosa **svih** prikažite **unutarnje čvorove** - varijabla **odg3**
- d) Nakon unosa **svih** prikažite **listove** - varijabla **odg4**

Primjer valjanog formata rješenja (nije točno rješenje zadatka)

```
odg1 = [1, 2, 3, 4, 5, 6]
odg2 = [3, 4, 5, 6, 7, 8]
odg3 = [1, 2, 9, 10, 11]
odg4 = [3, 4, 5, 6, 7]
```

Primjer

Za naredno AVL stablo:



i ove željene odgovore:

- **odg1** - Svi listovi
- **odg2** - Svi unutarnji čvorovi

dobivamo rješenje:

```
odg1 = [18, 40, 49]
odg2 = [34, 48]
# Poredak elemenata unutar nizova nije bitan
```

Napomena

- U zadatku se od vas traži da upisujete međurezultate. Pročitajte zadatak u cijelosti prije početka rješavanja jer bi se vaše stablo moglo bitno izmijeniti nakon dodavanja/brisanja dodatnih elemenata.
- U zadacima **ne smijete** dodavati dodatne, vlastite varijable, funkcije i sl.
 - Očekuju se samo tražene varijable (*odg1, odg2, ...*)
 - Definirajte sve varijable, makar prazne, u suprotnom vam mogu neki testovi padati!
 - Ako imate dodatnih varijabli, funkcija ili klasa, padat će vam svi testovi!
- Fotografije postupka ovog zadatka uploadate u odvojenom ispitu
 - Dobit ćete 0 bodova ako nemate uploadane fotografije postupka rješavanja ovog zadatka sa potpisanog papira

[AVL stablo](6 bodova)

U inicijalno prazno AVL stablo redom unesite sljedeće brojeve:

38, 19, 18, 26, 5, 3, 13, 8

Pitanja

- a) Nakon unosa **26** prikažite **listove** - varijabla **odg1**
- b) Nakon unosa **3** prikažite **unutarnje čvorove** - varijabla **odg2**
- c) Nakon unosa **svih** prikažite **unutarnje čvorove** - varijabla **odg3**
- d) Nakon unosa **svih** prikažite **listove** - varijabla **odg4**

Primjer valjanog formata rješenja (nije točno rješenje zadatka)

```
odg1 = [1, 2, 3, 4, 5, 6]
odg2 = [3, 4, 5, 6, 7, 8]
odg3 = [1, 2, 9, 10, 11]
odg4 = [3, 4, 5, 6, 7]
```

Primjer

Za naredno AVL stablo:



i ove željene odgovore:

- **odg1** - Svi listovi
- **odg2** - Svi unutarnji čvorovi

dobivamo rješenje:

```
odg1 = [18, 40, 49]
odg2 = [34, 48]
# Poredak elemenata unutar nizova nije bitan
```

Napomena

- U zadatku se od vas traži da upisujete međurezultate. Pročitajte zadatak u cijelosti prije početka rješavanja jer bi se vaše stablo moglo bitno izmijeniti nakon dodavanja/brisanja dodatnih elemenata.
- U zadacima **ne smijete** dodavati vlastite (dodatne) varijable, funkcije i sl.
 - Očekuju se samo tražene varijable (*odg1, odg2, ...*)
 - Definirajte sve varijable, makar prazne, u suprotnom vam mogu testovi padati!
 - Ako imate dodatnih, vlastitih varijabli, funkcija ili klasa, padat će vam svi testovi!
- Fotografije postupka ovog zadatka uploadate u odvojenom ispitu
 - Dobit ćete 0 bodova ako nemate uploadane fotografije postupka rješavanja ovog zadatka sa potpisanog papira

[6 bodova] Huffmanovo kodiranje

Pretpostavite izvorni alfabet $S=\{A,B,C,D,E\}$, vjerojatnosti $P=\{0.11, 0.22, 0.14, 0.25, 0.28\}$

- a) Izgradite Huffmanovu kodnu tablicu
- b) Enkodirajte ulazni niz BCECBA i izračunajte faktor kompresije (koja uzima u obzir veličine enkodiranog niza i kodne tablice)
- c) Dekodirajte dolazni tok 00011100000

U postupku izgradnje stabla, lijevo dijete uvijek neka bude ono manje, a desno neka bude veće (pogledajte **napomenu 1**). Ovo će osigurati jednoznačnost koda.

Nakon rješavanja unesite svoje rješenje:

- kodnu tablicu iz a) dijela zadatka kao varijablu **table** koja je riječnik (dictionary) koja kao ključeve sadrži simbole, a kao vrijednosti njihove pripadajuće kodne riječi
- enkodirani niz iz b) dijela zadatka kao varijablu **encoded** koja je string
- faktor kompresije iz b) dijela zadatka kao varijablu **R** koja je float
- dekodirani niz iz c) dijela zadatka kao varijablu **expanded** koja je string

Numerička tolerancija je 10^{-2} .

Primjer valjanog formata unosa rješenja (nije točno rješenje zadatka, npr. tablica čak nije ni prefix-free kod):

```
table={'A':'0', 'B':'01', 'C':'10', 'D':'111', 'E':'110'}

encoded='011110111011'

R=1.4

expanded='ACCCC'
```

Napomena 1 u postupku izgradnje kodne tablice, koristite sljedeći **tie-breaker** (razrješavanje situacija kada imate izjednačene usporedbe)...

Uvedimo imenovanje čvora u Huffmanovom stablu na sljedeći način:

- ako je list, tada mu je ime njegov pripadni simbol
- ako je unutarnji čvor, neka mu je ime uzlazno sortirana konkatenacija simbola njegovih listova. Npr. ako su u podstablu tog čvora listovi simbola 'C', 'A', 'E' onda će ime tog unutarnjeg čvora biti 'ACE'.

Ako dva čvora imaju jednake vjerojatnosti, tada se manjim čvorom smatra onaj koji ima leksikografski manje ime. Primjer istinitih usporedbi su: 'AB' < 'AC' < 'ACE'.

Napomena 2

- U zadacima **ne smijete** dodavati vlastite dodatne varijable, funkcije i sl.
 - Očekuju se samo tražene varijable
 - Morate definirati sve varijable, makar prazne, u suprotnom će vam padati testovi!
 - Ako imate dodatnih varijabli, funkcija ili klasa, padat će vam svi testovi!
- Fotografije postupka ovog zadatka uploadate u odvojenom ispitu
 - Dobit ćete 0 bodova ako nemate uploadane fotografije postupka rješavanja ovog zadatka sa potpisanog papira

[6 bodova] Huffmanovo kodiranje

Pretpostavite izvorni alfabet $S=\{A,B,C,D,E\}$, vjerojatnosti $P=\{0.13, 0.17, 0.23, 0.19, 0.28\}$

- a) Izgradite Huffmanovu kodnu tablicu
- b) Enkodirajte ulazni niz ABABBB i izračunajte faktor kompresije (koja uzima u obzir veličine enkodiranog niza i kodne tablice)
- c) Dekodirajte dolazni tok 11011011001110

U postupku izgradnje stabla, lijevo dijete uvijek neka bude ono manje, a desno neka bude veće (pogledajte **napomenu 1**). Ovo će osigurati jednoznačnost koda.

Nakon rješavanja unesite svoje rješenje:

- kodnu tablicu iz a) dijela zadatka kao varijablu **table** koja je riječnik (dictionary) koja kao ključeve sadrži simbole, a kao vrijednosti njihove pripadajuće kodne riječi
- enkodirani niz iz b) dijela zadatka kao varijablu **encoded** koja je string
- faktor kompresije iz b) dijela zadatka kao varijablu **R** koja je float
- dekodirani niz iz c) dijela zadatka kao varijablu **expanded** koja je string

Numerička tolerancija je 10^{-2} .

Primjer valjanog formata unosa rješenja (nije točno rješenje zadatka, npr. tablica čak nije ni prefix-free kod):

```
table={'A':'0', 'B':'01', 'C':'10', 'D':'111', 'E':'110'}

encoded='011110111011'

R=1.4

expanded='ACCCC'
```

Napomena 1 u postupku izgradnje kodne tablice, koristite sljedeći **tie-breaker** (razrješavanje situacija kada imate izjednačene usporedbe)...

Uvedimo imenovanje čvora u Huffmanovom stablu na sljedeći način:

- ako je list, tada mu je ime njegov pripadni simbol
- ako je unutarnji čvor, neka mu je ime uzlazno sortirana konkatenacija simbola njegovih listova. Npr. ako su u podstablu tog čvora listovi simbola 'C', 'A', 'E' onda će ime tog unutarnjeg čvora biti 'ACE'.

Ako dva čvora imaju jednake vjerojatnosti, tada se manjim čvorom smatra onaj koji ima leksikografski manje ime. Primjer istinitih usporedbi su: 'AB' < 'AC' < 'ACE'.

Napomena 2

- U zadacima **ne smijete** dodavati vlastite dodatne varijable, funkcije i sl.
 - Očekuju se samo tražene varijable
 - Morate definirati sve varijable, makar prazne, u suprotnom će vam padati testovi!
 - Ako imate dodatnih varijabli, funkcija ili klasa, padat će vam svi testovi!
- Fotografije postupka ovog zadatka uploadate u odvojenom ispitu
 - Dobit ćete 0 bodova ako nemate uploadane fotografije postupka rješavanja ovog zadatka sa potpisanog papira

[6 bodova] Huffmanovo kodiranje

Pretpostavite izvorni alfabet $S=\{A,B,C,D,E\}$, vjerojatnosti $P=\{0.31, 0.26, 0.17, 0.02, 0.24\}$

- a) Izgradite Huffmanovu kodnu tablicu
- b) Enkodirajte ulazni niz EACBAA i izračunajte faktor kompresije (koja uzima u obzir veličine enkodiranog niza i kodne tablice)
- c) Dekodirajte dolazni tok 110010000111

U postupku izgradnje stabla, lijevo dijete uvijek neka bude ono manje, a desno neka bude veće (pogledajte **napomenu 1**). Ovo će osigurati jednoznačnost koda.

Nakon rješavanja unesite svoje rješenje:

- kodnu tablicu iz a) dijela zadatka kao varijablu **table** koja je riječnik (dictionary) koja kao ključeve sadrži simbole, a kao vrijednosti njihove pripadajuće kodne riječi
- enkodirani niz iz b) dijela zadatka kao varijablu **encoded** koja je string
- faktor kompresije iz b) dijela zadatka kao varijablu **R** koja je float
- dekodirani niz iz c) dijela zadatka kao varijablu **expanded** koja je string

Numerička tolerancija je 10^{-2} .

Primjer valjanog formata unosa rješenja (nije točno rješenje zadatka, npr. tablica čak nije ni prefix-free kod):

```
table={'A':'0', 'B':'01', 'C':'10', 'D':'111', 'E':'110'}

encoded='011110111011'

R=1.4

expanded='ACCCC'
```

Napomena 1 u postupku izgradnje kodne tablice, koristite sljedeći **tie-breaker** (razrješavanje situacija kada imate izjednačene usporedbe)...

Uvedimo imenovanje čvora u Huffmanovom stablu na sljedeći način:

- ako je list, tada mu je ime njegov pripadni simbol
- ako je unutarnji čvor, neka mu je ime uzlazno sortirana konkatencija simbola njegovih listova. Npr. ako su u podstablu tog čvora listovi simbola 'C', 'A', 'E' onda će ime tog unutarnjeg čvora biti 'ACE'.

Ako dva čvora imaju jednake vjerojatnosti, tada se manjim čvorom smatra onaj koji ima leksikografski manje ime. Primjer istinitih usporedbi su: 'AB' < 'AC' < 'ACE'.

Napomena 2

- U zadacima **ne smijete** dodavati vlastite dodatne varijable, funkcije i sl.
 - Očekuju se samo tražene varijable
 - Morate definirati sve varijable, makar prazne, u suprotnom će vam padati testovi!
 - Ako imate dodatnih varijabli, funkcija ili klasa, padat će vam svi testovi!
- Fotografije postupka ovog zadatka uploadate u odvojenom ispitu
 - Dobit ćete 0 bodova ako nemate uploadane fotografije postupka rješavanja ovog zadatka sa potpisanog papira

[Trie](6 bodova)

U *trie* redom unesite znakovne nizove:

zzih, swmwz, zzhq, swmwzgg, zzihg, cswmwzgg

Zatim, nakon unosa redom izbrišite znakovne nizove:

swmwzgg, swmwz

Na papiru označite krajeve znakovnih nizova sa '#', ali kod unošenja samih znakovnih nizova u Edgar, **unosite bez oznake kraja niza!** Svaki odgovor od vas očekuje listu znakova, koje unosite kao Python liste *charactera*. **Redoslijed elemenata u tim listama nije bitan i neće utjecati na bodovanje.**

Krajnji čvorovi oni koji označavaju kraj znakovnih nizova, a **prijelazni čvorovi** svi ostali. Korijen ne spada niti u prijelazne, niti u krajnje čvorove, pa ga ne unosite nigdje.

Pitanja

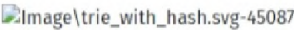
- a) Nakon unosa "swmwzgg" prikažite **krajnje čvorove** - varijabla **odg1**
- b) Nakon unosa **svih** prikažite **prijelazne čvorove** - varijabla **odg2**
- c) Nakon brisanja "swmwzgg" i "swmwz" prikažite **prijelazne čvorove** - varijabla **odg3**

Primjer valjanog formata rješenja (nije točno rješenje zadatka)

```
odg1 = ['a', 'b', 'c']
odg2 = ['d', 'e', 'f', 'g']
odg3 = ['i', 'j', 'k']
```

Primjer unosa

Za sljedeće *trie* stablo



i ove željene odgovore:

- **odg1** - Svi krajnji čvorovi
- **odg2** - Svi prijelazni čvorovi

dobivamo rješenje:

```
odg1 = ['z', 'i', 'o', 'o']
odg2 = ['l', 'a']
# Poredak elemenata unutar lista nije bitan
```

Napomena

- U zadatku se od vas traži da upisujete međurezultate. Pročitajte zadatak u cijelosti prije početka rješavanja jer bi se vaše stablo moglo bitno izmjeniti nakon dodavanja/brisanja dodatnih elemenata.
- U zadacima **ne smijete** dodavati vlastite (dodatne) varijable, funkcije i sl.
 - Očekuju se samo tražene varijable (*odg1, odg2, ...*)
 - Definirajte sve varijable, makar prazne, u suprotnom vam mogu testovi padati!
 - Ako imate dodatnih, vlastitih varijabli, funkcija ili klasa, padat će vam svi testovi!
- Fotografije postupka ovog zadatka uploadate u odvojenom ispitu
 - Dobit ćete 0 bodova ako nemate uploadane fotografije postupka rješavanja ovog zadatka sa potpisanog papira

[Trie](6 bodova)

U *trie* redom unesite znakovne nizove:

lmshry, lmw, fcjiuo, lmshryf, lmwmg, dlmw

Zatim, nakon unosa redom izbrišite znakovne nizove:

lmwmg, lmshryf

Na papiru označite krajeve znakovnih nizova sa '#', ali kod unošenja samih znakovnih nizova u Edgar, **unosite bez oznake kraja niza!** Svaki odgovor od vas očekuje listu znakova, koje unosite kao Python liste. **Redoslijed elemenata u tim listama nije bitan i neće utjecati na bodovanje.**

Krajnji čvorovi su oni koji označavaju kraj znakovnih nizova, a **prijelazni čvorovi** svi ostali. Korijen ne spada niti u prijelazne, niti u krajnje čvorove, pa ga ne trebate unositi.

Pitanja

- a) Nakon unosa "lmwmg" prikažite **krajnje čvorove** - varijabla *odg1*
- b) Nakon unosa *svih* prikažite **prijelazne čvorove** - varijabla *odg2*
- c) Nakon brisanja "lmwmg" i "lmshryf" prikažite **prijelazne čvorove** - varijabla *odg3*

Primjer valjanog formata rješenja (nije točno rješenje zadatka)

```
odg1 = ['a', 'b', 'c']
odg2 = ['d', 'e', 'f', 'g']
odg3 = ['i', 'j', 'k']
```

Primjer

Za sljedeće *trie* stablo



i ove željene odgovore:

- **odg1** - Svi krajnji čvorovi
- **odg2** - Svi prijelazni čvorovi

dobivamo rješenje:

```
odg1 = ['z', 'i', 'o', 'o']
odg2 = ['l', 'a']
# Poredak elemenata unutar lista nije bitan
```

Napomena

- U zadatku se od vas traži da upisujete međurezultate. Pročitajte zadatak u cijelosti prije početka rješavanja jer bi se vaše stablo moglo bitno izmjeniti nakon dodavanja/brisanja dodatnih elemenata.
- U zadacima **ne smijete** dodavati vlastite (dodatne) varijable, funkcije i sl.
 - Očekuju se samo tražene varijable (*odg1, odg2, ...*)
 - Definirajte sve varijable, makar prazne, u suprotnom vam mogu testovi padati!
 - Ako imate dodatnih, vlastitih varijabli, funkcija ili klasa, padat će vam svi testovi!
- Fotografije postupka ovog zadatka uploadate u odvojenom ispitu
 - Dobit ćete 0 bodova ako nemate uploadane fotografije postupka rješavanja ovog zadatka sa potpisanog papira

[Trie](6 bodova)

U *trie* redom unesite znakovne nizove:

vfur, vfbwx, qvqdz, vfbwxj, vfurxnj, ievfur

Zatim, nakon unosa redom izbrišite znakovne nizove:

vfbwx, vfurxnj

Na papiru označite krajeve znakovnih nizova sa '#', ali kod unošenja samih znakovnih nizova u Edgar, **unosite bez oznake kraja niza!** Svaki odgovor od vas očekuje listu znakova, koje unosite kao Python liste *charactera*. **Redoslijed elemenata u tim listama nije bitan i neće utjecati na bodovanje.**

Krajnji čvorovi oni koji označavaju kraj znakovnih nizova, a **prijelazni čvorovi** svi ostali. Korijen ne spada niti u prijelazne, niti u krajnje čvorove, pa ga ne unosite nigdje.

Pitanja

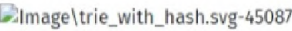
- a) Nakon unosa "vfbwxj" prikažite **krajnje čvorove** - varijabla **odg1**
- b) Nakon unosa **svih** prikažite **prijelazne čvorove** - varijabla **odg2**
- c) Nakon brisanja "vfbwx" i "vfurxnj" prikažite **prijelazne čvorove** - varijabla **odg3**

Primjer valjanog formata rješenja (nije točno rješenje zadatka)

```
odg1 = ['a', 'b', 'c']
odg2 = ['d', 'e', 'f', 'g']
odg3 = ['i', 'j', 'k']
```

Primjer unosa

Za sljedeće *trie* stablo



i ove željene odgovore:

- **odg1** - Svi krajnji čvorovi
- **odg2** - Svi prijelazni čvorovi

dobivamo rješenje:

```
odg1 = ['z', 'i', 'o', 'o']
odg2 = ['l', 'a']
# Poredak elemenata unutar lista nije bitan
```

Napomena

- U zadatku se od vas traži da upisujete međurezultate. Pročitajte zadatak u cijelosti prije početka rješavanja jer bi se vaše stablo moglo bitno izmijeniti nakon dodavanja/brisanja dodatnih elemenata.
- U zadacima **ne smijete** dodavati vlastite (dodatne) varijable, funkcije i sl.
 - Očekuju se samo tražene varijable (*odg1, odg2, ...*)
 - Definirajte sve varijable, makar prazne, u suprotnom vam mogu testovi padati!
 - Ako imate dodatnih, vlastitih varijabli, funkcija ili klasa, padat će vam svi testovi!
- Fotografije postupka ovog zadatka uploadate u odvojenom ispitu
 - Dobit ćete 0 bodova ako nemate uploadane fotografije postupka rješavanja ovog zadatka sa potpisanog papira