

[6 bodova] TEŽI ZADATAK

Imate ulazni niz koji se sastoji od 33 uzastopna ponavljanja niza "ab". Komprimirajte taj niz algoritmom LZ77 na najmanju moguću veličinu. Pritom, imate na početku slobodu biranja hiperparametara algoritma (veličina rječnika i veličina spremnika za pretraživanje). U slučaju da postoji više mogućih rješenja identične duljine, odaberite onu varijantu koja ima minimalno zauzeće radne memorije pri dekompresiji (dekodiranju).

Kao rješenje unesite:

- komprimirani izlaz kao varijablu **output** koji je n-torka (tuple) sa ugniježđenim n-torkama po potrebi (kako smo radili na predavanjima)
- duljinu ulaznog niza znakova u bitovima kao varijablu **input_length** koja sadrži cijeli broj (pretpostavljajući ASCII kodnu tablicu za znakove)
- duljinu izlaznog niza znakova u bitovima kao varijablu **output_length** koja sadrži cijeli broj. Uzmite u obzir da se za svaki element koristi najmanji broj bitova potreban za pohranu Vaših podataka (određeno hiperparametrima).
- faktor kompresije kao varijablu **compression_factor** koja sadrži decimalni broj.
- odabrani hiperparametar veličina rječnika kao varijablu **Ld** koja sadrži cijeli broj
- odabrani hiperparametar veličina spremnika za pretraživanje kao varijablu **La** koja sadrži cijeli broj

Numerička tolerancija je 10^{-2} .

Primjer valjanog formata unosa rješenja (nije točno rješenje zadatka, čak niti veličine dimenzija, a primjer outputa je samo ilustrativan i nije sintaksno točan - morate koristiti sintaksu sa predavanja):

```
output = (("a", "bc", 8, -1), "a", (0, 8, "aaaa"))
```

```
input_length = 18
```

```
output_length = 10
```

```
compression_factor = 0.77
```

```
Ld = 12
```

```
La = 1024
```

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

$$\max -3x_1 + 1x_2 + 5x_3$$

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

$$3x_1 + x_2 - 6x_3 \leq 0.8$$

$$3x_1 + 2x_2 \leq 3$$

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

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):

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, upload je opcionalan

Student's answer:

```
1 status = 1
2
3 tableau = [
4 [2.625,0,0,1.25,0,1.125,4.625],
5 [-0.375,0,1,0.25,0,0.125,0.625],
6 [-0.75,0,0,1.5,1,0.25,3.05],
7 [1.5,1,0,0,0,0.5,1.5]
8 ]
9
10 solution = [0,1.5,0.625,0,3.05,0]
11
12 obj = -4.625
13
```

Correct answer:

```
1 This exam does not have "show solutions" option enabled.
```

[B-stablo](6 bodova)

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

64, 5, 55, 50, 1, 45, 65, 43, 62, 9, 77, 19, 20, 73, 46

Zatim redom izbrišite:

19, 62, 43, 1

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 **9** prikažite **listove** - varijabla **odg1**
- c) Nakon unosa **svih** prikažite **unutarnje čvorove** - varijabla **odg2**
- b) Nakon brisanja **19, 62, 43 i 1** prikažite **listove** - varijabla **odg3**

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]]
```

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, ...*)

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 (ako želite)

Student's answer:

```
1 odg1=[[1],[9,43,45],[55],[64,65]]
2 odg2=[[19,50],[5],[43],[62,65]]
3 odg3=[[5,9],[45,46],[55,64],[73,77]]
4
5
6
7
8
9
10
11
12
13
```

Correct answer:

```
1 This exam does not have "show solutions" option enabled.
```

Hint: Correct. Well done!

[Patricia Trie](6 bodova)

U *Patricia trie* redom unesite znakovne nizove:

lmsx, lmswxi, xiuo, lmswhry, lmswmxg, xdamw

Zatim, nakon unosa redom izbrišite znakovne nizove:

xdamw, lmswxi

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 "**lmswhry**" prikažite **krajnje čvorove** - varijabla **odg1**
- b) Nakon unosa **svih** prikažite **prijelazne čvorove** - varijabla **odg2**
- c) Nakon brisanja "**xdamw**" i "**lmswxi**" prikažite **krajnje čvorove** - varijabla **odg3**

Primjer valjanog formata rješenja (nije točno rješenje zadatka)

```
odg1 = ['abc', 'b', 'cd']
odg2 = ['de', 'ef', 'g']
odg3 = ['ijk', 'jk', 'k']
```

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

Student's answer:

```
1 odg1=['x','xi','hry','xiuo']
2 odg2=['x','lms','w']
3 odg3=['x','xiuo','mxg','hry']
4
5
6
7
8
9
10
11
12
13
```

Correct answer:

```
1 This exam does not have "show solutions" option enabled.
```

Hint: Correct. Well done!

[6 bodova] Stablo prioriteta

Izgradite stablo prioriteta za upita oblika: $[x_1, x_2] \times [y, \infty)$, nad sljedećim točkama:

$$\begin{aligned} &(-22, 22), (-4, 22), (-6, 10), (1, 10), \\ &(24, 6), (-6, 3), (-18, 3), (14, 1), \\ &(-8, -10), (-9, -14), (-25, -20), (-25, -29) \end{aligned}$$

- **Provjereni** čvorovi su svi za koje **moramo** ispitati pripadnost $[x_1, x_2]$ i $[y, \infty)$ kako bi znali jesu li upali u rezultat. Gledajte kao da se provjere za x i y vrijednosti odvijaju istovremeno.
- **Poluprovjereni** čvorovi su svi za koje **moramo** ispitati samo pripadnost intervalu $[y, \infty)$ kako bi znali jesu li upali u rezultat.

PAZITE!!! - Za neke čvorove znamo da sigurno ne ulaze u rezultat i oni ne ulaze ni u provjerene ni poluprovjerene.

Pitanja

- a) Zapišite vrijednosti u podstrukturi binarnog stabla za pretraživanje sadržanog u stablu prioriteta u *varijablu* **bst** (poredak nije bitan).
- b) Izvršite upit za vrijednosti u intervalu $[2, 18] \times [-25, \infty)$ i zapišite **provjerene** čvorove (točke) u *varijablu* **visited1**
- c) Izvršite upit za vrijednosti u intervalu $[-24, 13] \times [-4, \infty)$ i zapišite **poluprovjerene** čvorove (točke) u *varijablu* **pruned2**
- d) Izvršite upit za vrijednosti u intervalu $[-14, 14] \times [0, \infty)$ i zapišite **provjerene** čvorove (točke) u *varijablu* **visited3**

Primjer rezultata:

```
bst = [1, 4, 2, 3, 10]
visited1 = [(1, 2), (1, 3), (2, 4)] # Točke koje se nalaze u `provjerenim` čvorovima
pruned2 = [(1, 2), (1, 3), (2, 4)] # Točke koje se nalaze u `poluprovjerenim` čvorovima
visited3 = [(1, 2), (1, 3), (2, 4)] # Točke koje se nalaze u `provjerenim` čvorovima
```