

Skilaverkefni03. Stærð- og Reiknifræði REI201G

Donn Eunice Bartido deb5@hi.is

Janúar 2023

Verkefni 12: Shell-röðun

Skrifið Python-fall sem Shell-raðar. Byrjið á að leysa verkefni 9 ef þið eruð ekki þegar búin að því. Það er hægt að nota heiltöludeilingu (//) til að reikna bil og hlutrunan í neðstu línunni fæst með `a[i:N:bil]`. Byggið á reikniritinu að framan (og alls ekki ná í tilbúið forrit af netinu). Prófið að raða listunum `[8,3,2]` og `[8,5,1,9,6,2,1,7,11,3]`, og auk þess einum sjálfvöldum lista með 12 tveggja stafa tölum.

Lausn

```
In [2]: def insertsort(A):
        i = 1
        while i < len(A):
            j = i
            while j > 0 and A[j-1] > A[j]:
                A[j-1], A[j] = A[j], A[j-1]
                j -= 1
            i += 1
        return A

def shellsort(a):
    k = 1
    N = len(a)
    while(True):
        bil = N//2**k
        k+= 1
        if bil < 1:
            break
        for i in range(bil):
            b = insertsort(a[i:N:bil])
        return b

print(shellsort([8,3,2]))
print(shellsort([8,5,1,9,6,2,1,7,11,3]))
print(shellsort([67, 92, 18, 44, 67, 71, 77, 58, 29, 40, 11, 13]))

[2, 3, 8]
[1, 1, 2, 3, 5, 6, 7, 8, 9, 11]
[11, 13, 18, 29, 40, 44, 58, 67, 67, 71, 77, 92]
```

Verkefni 13: Skrá með íslenskum orðum

Þetta verkefni er lauslega byggt á verkefnum í 9. kafla Think Python kennslubókarinnar. Í skrá með veffang <https://cs.hi.is/python/ord.txt> eru 217 þúsund íslensk orð. Þið getið

skoðað skrána með því að smella á hlekkinn. Í verkefninu sem hér fylgir þarf að nota ýmsar strengjaaðgerðir sem lýst er í kafla 5.2.

1. Opnið skrána með urlopen og lesið hana inn eins og sýnt er í kafla 8.2.4. Prentið út fyrstu 5 orðin og líka tíu þúsundasta hvert orð og öll orð sem eru lengri en 30 stafir.

Leiðbeining

Það er bara eitt orð á hverri línu sem hægt er að ná í með `ord = lína.decode().strip()`

Athugasemd

Í stað þess að nota `urllib` mætti lesa skrána inn með `NumPy` eins og fjallað er um í kafla 10.6.3. Þá er notað `dtype=str` í sjálfum innlestinum, og síðan má nota `tolist`-aðferðina til að breyta innlesnum vigri í venjulegan lista af strengjum.

1. Spegilorð (palindrome) er orð sem er eins lesið afturábak og áfram (t.d. `kajak`).
Skrifið rökfall `spgilorð(s)` sem kannar hvort `s` sé spegilorð [rökfall er fall sem skilar `True` eða `False` og prófið. Skrifið í framhaldi út öll spegilorð í skránni, 10 á hverja línu. Fyrsta linan ætti að verða:

abba, afa, aga, agga, aka, ala, alla, ama, amma, ana

1. Finnið þau orð í skránni sem hafa einn sérhljóða og hámarksfjölda samhljóða

Lausn

1)

```
In [6]: from urllib.request import urlopen
f = urlopen("https://cs.hi.is/python/ord.txt")
ordList = []

for line in f:
    ordList.append(line.decode().strip())

fimmOrd = ""
for i in range(5):
    fimmOrd += ordList[i] + " "

print(fimmOrd)

for k,i in enumerate(ordList, 1):
    if k % 10000 == 0 or len(i) > 30:
        print(f'{k:6}', i)
```

abba abbadís abbadísar abbadísarinnar abbadísartíð
 3512 alþjóðaheilbrigðisstofnunarinnar
 3574 alþjóðasiglingamálastofnunarinnar
 5822 atvinnuleysistryggingasjóðurinn
 10000 barónessunni
 20000 bókmenntaheimurinn
 30000 eldvarpa
 39136 flugslysarannsóknarnefndarinnar
 40000 flögrað
 50000 galdrakerlingin
 60000 hafnarverkamannsins
 70000 hnýttu
 80000 illkvittnislega
 90000 konunglegan
 100000 leiðbeiningu
 110000 margnefndi
 120000 nemandi
 121142 norðuratlantshafssjávarspendýraráðsins
 121175 norðurheimskautsrannsóknaráðsins
 130000 ramman
 140000 sandhólum
 150000 skynsemd
 160000 stjórnarþátttöku
 170000 sólarhofsins
 172917 teiknimyndævintýrapoppálfrkonan
 180000 tötralegum
 190000 veðurratsjá
 200000 árásargjörn
 210000 útdauðar

2)

```
In [10]: def palindrome(s):
          return (s==s[::-1])

count = 0

for i in ordList:
    if palindrome(i):
        if count == 10:
            print()
            count = 0
        else:
            print(i, end=" ", " ")
            count +=1
```

abba, afa, aga, agga, aka, ala, alla, ama, amma, ana,
 argra, assa, ata, axa, aða, gíg, gýg, illi, inni, iðaði,
 kajak, kok, kák, kæk, kók, kúk, mm, muninum, munnum, munum,
 mussum, natan, nón, píp, rabbar, radar, raddar, rafar, ragar, rakar,
 rammar, rappar, rasar, rassar, ratar, raðar, rifir, riðir, ruddur, rullur,
 runur, rár, rær, rór, rör, rýr, sinnis, stúts, summus, sás,
 tillit, tætt, uku, ullu, undnu, unnu, unu, uxu, á, æ,
 ísí, ó, óbó, ódó, óró, ý,

3)

```
In [11]: def longest(s):
          vowels = ['a', 'á', 'e', 'é', 'i', 'í', 'o', 'ó', 'u', 'ú', 'y', 'ý', 'æ']
          longword = []
          longest_word_length = 0
          longest_words = []
          for i in s:
```

```

word = i.lower()
count = sum(1 for char in word if char in vowels)
if count == 1:
    longword.append(word)
    word_length = len(word)
    if word_length > longest_word_length:
        longest_word_length = word_length
for word in longword:
    if len(word) == longest_word_length:
        longest_words.append(word)
print(longest_words)
longest(ordList)

```

```
['bhmfólks', 'skrappst', 'skyggnst', 'strengst']
```

Verkefni 15: Sameining nafnaskrár og einkunnaskrár

Útbúna hafa verið tvær skrár með gervigögnum: Sú fyrri,

<https://cs.hi.is/python/einkunn.txt> inniheldur prófnúmer og einkunnir og sú seinni,

<https://cs.hi.is/python/nofn.txt>, inniheldur prófnúmer og nöfn.

1. **Skrá lesin.** Skriðið forrit sem les einkunnaskrána inn í uppflittitöflu (dictionary) eink_tafla þannig að eink_tafla[nr] gefi einkunn próftaka með prófnúmer nr sbr. dæmin í kafla 8.2.5. Prófið (t.d. með því að sýna úttakið sem skipunin print(eink_tafla) skilar).

Uppflittitafla. Skráin nofn.txt byrjar svona:

0176 Kjartan Valur Jónsson 0542 Aðalheiður Pétursdóttir 0970 Sigrún Ása Jónsdóttir
 Prófnúmerið er sem sé aðskilið frá nafninu með tveimur bilum og svo er nafnið ýmist með millinafni eða án. Til að höndla þetta er hægt að nota split með viðbótarviðfangi sem tilgreinir hve oft á að skipta. Ef lína er fyrsta línan í skránni mundi

```
(nr,nafn) = lína.decode().strip().split(maxsplit=1)
```

skipta strengnum rétt í tvo hluta, númer og nafn (strip()) þarf að vera með hér til að losna við línuskiptatákn). Notið þetta trix til að lesa nafnaskrána inn í uppflittitöflu nafn_tafla sem nota má til að fletta upp á nafni útfrá prófnúmeri. Prófið með sama hætti og fyrr.

1. **Úskrift lista.** Skriðið forrit sem rennir í gegn um prófnúmerin í eink_tafla, flettir upp nafni hvers nemanda í nafn_tafla, og skrifar allt þrennt: prófnúmer, einkunn, og nafn í snyrtilegri töflu með fyrirsögnum og dálkum sem standast á.
2. **Hæsta einkunn.** Skriðið loks forrit sem ákvarðar hver fékk hæstu einkunnina og skrifar nafn hans/hennar ásamt einkunninni (gera má ráð fyrir að það sé bara einn). Notið lykkju og uppfærið inni í lykkjunni, í hvert sinn sem einkunn er hærri en sú hæsta sem sést hefur, bæði hæstu einkunnina og nafn þess sem hana fékk. Forritið ætti að skrifa: Íris María Birgisdóttir var hæst með 10.0).

In [9]: `### Lausn`

1) og 2)

```
In [16]: einkunnir = {}
f = urlopen("https://cs.hi.is/python/einkunn.txt")
for line in f:
    (tala, einkunn) = line.decode().strip().split()
    einkunnir[tala] = einkunn
print(einkunnir)

nofn = {}
f = urlopen("https://cs.hi.is/python/nofn.txt")
for line in f:
    (tala,nafn) = line.decode().strip().split(maxsplit=1)
    nofn[tala] = nafn
print(nofn)
```

```
{'0176': '7.0', '0542': '8.0', '0970': '9.5', '1419': '6.5', '1577': '7.0',
'2785': '9.0', '4218': '7.0', '4854': '9.0', '5469': '10.0', '6324': '6.5',
'6558': '8.5', '7923': '5.5', '8003': '9.5', '8148': '5.0', '8492': '7.5',
'9058': '7.5', '9134': '8.0', '9158': '8.0', '9340': '6.5', '9572': '8.0',
'9576': '5.5', '9595': '8.5', '9649': '6.5', '9706': '9.0'}
{'0176': 'Kjartan Valur Jónsson', '0542': 'Aðalheiður Pétursdóttir', '0970':
'Sigrún Ása Jónsdóttir', '1419': 'Erla Ýr Guðnadóttir', '1577': 'Hulda Ósk J
ónasdóttir', '2785': 'Bjarki Már Sveinsson', '4218': 'Kristín Fjóludóttir',
'4854': 'Anton Ingi Þórsson', '5469': 'Íris María Birgisdóttir', '6324': 'Ív
ar Sigurðsson', '6558': 'Ágúst Guðni Ingason', '7923': 'Steinunn Guðlaug Gun
narsdóttir', '8003': 'Eydís Þorsteinsdóttir', '8148': 'Signý Guðrún Pálsdótt
ir', '8492': 'Andri Oddur Steinarsson', '9058': 'Elías Ari Heimisson', '913
4': 'Birta Lárusdóttir', '9158': 'Anna Sveinbjörnsdóttir', '9340': 'Bjarni R
únar Kjartansson', '9572': 'Arna Þórisdóttir', '9576': 'Mark Johnson', '959
5': 'Víðir Kristjánsson', '9649': 'Anna Kristinsdóttir', '9706': 'Jónas Vald
imarsson'}
```

3)

```
In [17]: for i in einkunnir:
          print(i,einkunnir[i], nofn[i])
```

```
0176 7.0 Kjartan Valur Jónsson
0542 8.0 Aðalheiður Pétursdóttir
0970 9.5 Sigrún Ása Jónsdóttir
1419 6.5 Erla Ýr Guðnadóttir
1577 7.0 Hulda Ósk Jónasdóttir
2785 9.0 Bjarki Már Sveinsson
4218 7.0 Kristín Fjóludóttir
4854 9.0 Anton Ingi Þórsson
5469 10.0 Íris María Birgisdóttir
6324 6.5 Ívar Sigurðsson
6558 8.5 Ágúst Guðni Ingason
7923 5.5 Steinunn Guðlaug Gunnarsdóttir
8003 9.5 Eydís Þorsteinsdóttir
8148 5.0 Signý Guðrún Pálsdóttir
8492 7.5 Andri Oddur Steinarsson
9058 7.5 Elías Ari Heimisson
9134 8.0 Birta Lárusdóttir
9158 8.0 Anna Sveinbjörnsdóttir
9340 6.5 Bjarni Rúnar Kjartansson
9572 8.0 Arna Þórisdóttir
9576 5.5 Mark Johnson
9595 8.5 Víðir Kristjánsson
9649 6.5 Anna Kristinsdóttir
9706 9.0 Jónas Valdimarsson
```

4)

```
In [18]: def highest(eink,nafn):  
         highestGrade = 0  
         highestName = 0  
         for i in eink:  
             if (float(eink[i]) > highestGrade):  
                 highestGrade = float(eink[i])  
                 highestName = i  
         print(f'{nafn[highestName]} var með hæstu einkunn: {highestGrade}')  
  
highest(einkunnir,nofn)
```

Íris María Birgisdóttir var með hæstu einkunn: 10.0