

Увод у рачунарство 1

Александар Картељ
kartelj@matf.bg.ac.rs

Фундаментални алгоритми

Упоредивање, претраживање, сортирање,
генерисање комбинаторних објеката, ...

Фундаментални алгоритми

- Размотрићемо наредне стандардне проблеме и њихова решења:
 - Поређење објеката
 - Претраживање листи
 - Сортирање листи
 - Генерисање комбинаторних објеката:
варијације, комбинације, пермутације

Поређење објеката

- Често је потребно упоредити објекте на једнакост
- У зависности од тога шта је смисао тих објеката, једнакост не мора увек значити и чисто синтаксно поклапање нпр. разломци $3/6$ и $1/2$ су једнаки иако записани текст није
- Две листе су једнаке ако су исте дужине и на истим позицијама имају исте вредности (чак иако су у питању разломци)
- Пример једнаких листи: $[1, 2, 3]$ и $[1, 2, 3]$
- Пример различитих листи: $[1, 2, 3]$ и $[3, 2, 1]$

Пример

- Напишите функцију која враћа да ли су две листе једнаке

nije dozvoljeno da samo stavite lista1==lista2

```
def jednake(lista1, lista2):  
    # ako su razlicite duzine, onda su sigurno razlicite liste  
    if len(lista1)!=len(lista2):  
        return False  
    for i in range(len(lista1)):  
        # ako na nekoj poziciji u listama elementi nisu isti  
        # onda sigurno ni liste nisu iste pa odmah izlazimo  
        if lista1[i]!=lista2[i]:  
            return False  
    # samim tim sto smo dosli do ove linije,  
    # proverili smo da su liste identicne, jer bismo izasli  
    # ranije u suprotnom  
    return True
```

Уређена n-торка

- За даљи рад биће нам потреба структура података која се зове уређена n-торка или само n-торка
- У програмском језику Python ова структура има следећу форму:
(вредност 1, вредност 2, ..., вредност M)
при чему вредности не морају бити истих типова
- Приметити да се n-торка разликује од листе:
 - Листа се састоји од нефиксираног броја елемената и сви елементи су истог типа
 - n-торка се састоји од фиксираног броја елемената који могу бити различитог типа
- Пример n-торке која описује студента: ("Petar", "Petrovic", 1995, 8.75)

Пример (2)

- Напишите функцију која враћа да ли су два разломка иста.
За представљање разломака користити n-торку

Решење је превелико за слајд, налази се у додатним материјалима.

Пример (3)

- Написати функцију која проверава да ли су две листе разломака исте

Искомбиновати решења за једнакост листи и једнакост разломака.

Претраживање листе

- Под претраживањем за дату листу елемената подразумевамо одређивање индекса елемента листе:
 - који је једнак датој вредности
 - или испуњава неко друго задато својство (нпр. највећи елемент низа)
- Како бисте у листи испод пронашли елемент 15:
[123, 2, 25, 67, 334, 15, 343, 223, 15, 67]
- Опишите алгоритам који би решио овај проблем

Пример (4)

- Напишите функцију која прихвата листу и елемент који се тражи, а враћа позицију првог појављивања тог елемента или -1 ако се елемент не налази у листи

```
def pretrazi_listu(lista, el):  
    for i in range(len(lista)):  
        if lista[i]==el:  
            return i  
  
    # ako smo stigli do ove linije koda, to znaci da  
    # nismo pronasli element u listi i vracamo tada  
    # specijalni indeks -1  
    return -1
```

Претраживање листе (2)

- Шта уколико је листа по којој претражујемо сортирана?
- Нпр. тражимо елемент 15 у листи:
[2, 15, 15, 25, 67, 67, 123, 223, 334, 343]
- Да ли у том случају можемо брже да пронађемо елемент?

Бинарна претрага

- Може се применити само над сортираном листом!

***Решење је превелико за слајд,
налази се у додатним материјалима.***

Пример (5)

- Напишите програм који погађа број из опсега од 1 до 1000 који сте замислили са највише 10 питања

Материјал за читање

- За боље разумевање фундаменталних алгоритама прочитати Главу 6 из <http://poincare.matf.bg.ac.rs/~janicic//courses/p2.pdf>