



Scripting Languages

Python – algorithm exercises

- caterpillar
- leader



Exercise 1

Find the number of contiguous subsequences with the given sum
(caterpillar algorithm)

$s=12$

answer=3

4	5	3	5	6	3	3	8	1	9
0	1	2	3	4	5	6	7	8	9
12				12			12		

- all numbers are positive integers $0 < n \leq 100$
- subsequences may overlap
- Read data (as lists) from pickle-files:
 - "caterpillar_1e1.pkl" ... 1e5, 1e6, 1e7 $\rightarrow n = 10, 100000, 1000000, 10000000$
- Find out the algorithm time complexity – relation between number of elements (n) and execution time t

Results for $s = 190$:

1e1 \rightarrow 1, 1e5 \rightarrow 2011, 1e6 \rightarrow 19990, 1e7 \rightarrow 199883

$t(n) \sim n$



Program structure

```
import pickle
import time

datafile = open("caterpillar_1e1.pkl", 'rb')
data = pickle.load(datafile)
datafile.close()

n = len(data)
s = 190
ans = 0

start = time.time()

here goes your code
...

t = time.time()-start
print(n, ans, t)
```

execution time
measurement



Exercise 2

Find the leader

- The leader is the candidate whose name occurs more than $n/2$ times in n -votes
- The list of candidates is unknown
- Sequence may have only one leader or none

candidates = 4 \rightarrow [3,4,5,6]

votes = 10

leader = "5"

4	5	3	5	6	5	5	5	3	5
0	1	2	3	4	5	6	7	8	9

- if the leader exists, the sequence may still contain up to $n/2$ different candidates
- the sequence of votes may be huge – both execution time and memory consumption are important factors



Exercise 2

- Solutions:
 - naive: identify all different candidates, count the votes for each candidate separately and confirm its leadership
 - $\text{time}(n) \sim n^2$
 - $\text{mem}(n) \sim n$
 - better: make the table of votes for all candidates, choose the candidate with highest vote number and confirm its leadership
 - $\text{time}(n) \sim n$
 - $\text{mem}(n) \sim n$
 - good: sort (in place) the table of votes, take the candidate name from the middle position ($n/2 \pm 1$), count its votes and confirm its leadership
 - $\text{time}(n) \sim n \cdot \log(n)$
 - $\text{mem}(n) \sim n$



Exercise 2

Find the best solution:

- $\text{time}(n) \sim n$
- $\text{mem}(n) \sim 1$
- read data (as lists) from pickle-files:
 - "leader_1e3.pkl" ... 1e5, 1e7
 - $n = 1000, 100000, 100000000$
- candidates: positive integers
- print leader value (or -1 if none)

Results:

1e3 → leader = 14 (537 votes)

1e5 → leader = 37820 (votes 50214)

1e7 → leader = 4740208 (5001628 votes)



Program structure

```
import pickle
import time

datafile = open("leader_1e1.pkl", 'rb')
data = pickle.load(datafile)
datafile.close()

n = len(data)
leader = -1

start = time.time()

here goes your code
...

t = time.time()-start
print(n, leader, t)
```