VE477
Introduction to Algorithms

Lab 1

Manuel — UM-JI (Fall 2018)

Goals of the lab

- Review on dictionaries
- Review on hash tables
- Starting with Python

1 C programming

Using the C standard library, write an efficient implementation of the following data structures:

- Dictionary with the seven basic operation. Use sorted double link lists.
- Implement hash tables.

2 Starting with Python

Python is an interpreted programming language first developed in the early 1990s. Being high-level and general purpose it is widely used in computer science and various fields of engineering.

Install Python 3 and run the `python3` command in a terminal. On Linux systems it is recommended to install software through the package manager.

Basic prompt

At the python prompt run the following basic commands:

```
print("Good morning!")
(1+2+3*2)/3\%5
a=0; b=a+3; b=3*b; print(b)
# print("Good morning!")
print(I said "Good morning!")
print("Good\
Morning!")
a="good"; b="morning"; print(a + " " + b + "!")
nb = input("Input a number: ")
help()
```

Arrays

Python has no native implementation for arrays but instead features *tuples*, *lists*, and *dictionaries*.

List: simple list of values; index starts at 0;

Tuples: similar to lists but read-only;

Dictionaries: values are not numbered, neither are they ordered; values can be accessed using keys;

Example

```python
# tuple
days = ('Monday','Tuesday','Wednesday','Thursday','Friday','Saturday', 'Sunday')
# list
students = ['Sam', 'Tom', 'Laura', 'Dave', 'Sarah']
print(students[2])
students.append('Catherine')
del students[3]
print(students[2:4])
# dictionary
gradebook = {'Andrew Parson': [ 'a-', 'b', 'c+'], \
  'Paul Black': [ 'a-', 'b+', 'a+' ]}
del gradebook['Andrew Parson']
gradebook['Angela Tree'] = ['d', 'c-', 'f']
gradebook
```

Loops and conditional statements

In the rest of the lab it is recommended to write the code is a text editor and then run it from the terminal using the command `python filename.py`.

In Python any code **must be indented**. A program that is not properly indented **will not run**.

If statement

```python
#!/usr/bin/python
a=0; b=1; c=4
if a < 3:
  print(a)
if a < 3 and b > 0:
  print (c)
elif b == -1:
  print(b)
else:
  print(a)
```

Ternary operator

```python
a=0; b=1
a if a < b else b
```

While loops

```python
#!/usr/bin/python
a=0
while a != 0:
  a = input("Input a number: ")
  a+=1
  print(a)
print('Bye')
```

For loops

```python
#!/usr/bin/python
for i in list(range(3,12,2)):
  print(i)
students = ['Sam', 'Tom', 'Laura', 'Dave', 'Sarah']
for i in students:
  print("{} is a student".format(i))
```

Defining and calling functions

Defining functions requires to use the keyword `def`. Python programs do not need to define a `main` function.

Simple functions

```python
def gm(course):
  return "Good morning " + course + "!"
course='ve477'
message=gm(course)
print(message)
print(gm(input("Input course code: ")))
def list(students,courses):
  for i in students:
    print(i)
    del students[1]
  courses.append('ve281')
  return students,courses
c = ['ve477', 've370']
s = ['Sam', 'Tom', 'Laura', 'Dave', 'Sarah']
a,b=list(s, c)
print(students)
print(a, b, d, e)
```

Function arguments

```python
def list(students,courses):
  for i in students:
    print(i)
    del students[1]
  courses.append('ve281')
return students,courses
c = ['ve477', 've370']
s = ['Sam', 'Tom', 'Laura', 'Dave', 'Sarah']
# the input order can be changed
d,e=list(courses=c,students=s)
# define a function with a variable number of input
def varin(*args):
  return args
varin(1,2,3); varin(1); varin(1,2,3,4,5,6,7,8)
# define a function with a variable number of keywords
def varkw(**kwargs):
  return kwargs
varkw(students=['Sam', 'Tom'], courses=['ve477', 've370'],\
  sports=['tennis', 'soccer'])
```

First class functions

In Python a function can be passed as an argument or returned by a function.

```python
def adding(x):
  def add(y):
    return x+y
  return add
add_2=adding(2)
print(add_2(123))
```

List, set, and dictionary comprehensions

A list, set, or dictionary comprehension is a way to construct new lists, sets, or dictionaries from existing ones. It follows a syntax similar to set definition in mathematics.

```python
[i+3 for i in [1, 2, 3]]
[5*i for i in [3, 4, 5, 6, 7] if i < 5]
{x for x in 'abcdef' if x not in 'abc'}
{i: i**3 for i in range(2,10,3)}
```

Lambda functions

A Lambda function is a Python facility allowing to embed a function within the code. It can be viewed as an "embedded function" which does not need to be declared using `def`.

Basic use

```python
l = [lambda x: x ** 2, lambda x: x ** 3, lambda x: x ** 4]
for i in l:
  print(i(2))
min = (lambda x, y: x if x < y else y)
min(12,16)
```

Lambda and map functions

```python
# distances in Miles
mi=[ 123, 2, 45, 87.2, 192 ]
# distances in km
km=map(lambda x: x*1.609344,mi)
for i in km:
  print(i)
```

Lambda and filter functions

```python
gradebook = {'Andrew Parson': [ 55, 40, 60], 'Paul Black': [ 85, 80, 95 ]}
list(filter(lambda x: x < 60, gradebook['Andrew Parson']))
```

Exercise

Python 3 short practice:

- Construct more complex examples showing how to use:
  - Lambda functions with `map()`, and `filter()`
  - List, set, and dictionary comprehensions
- Write a phone book application, where the user can add, remove, and edit entries.