

# Lab5: Algorithms

This lab is NOT part of your lab coursework but you are strongly recommend to try it. The material is examinable.

If you have questions about this lab sheet or are stuck with one of the questions talk to the tutors in the lab or post your question on Moodle. If your question needs to include code, please email it to the mailing list rather than posting it to Moodle.

## 1 Dictionaries

**Exercises 1.** Look at the following code which uses dictionaries. What does it do?

```
result = {  
    'a': lambda x: x * 5,  
    'b': lambda x: x + 7,  
    'c': lambda x: x - 2  
}[value](x)
```

*Hint: Type it in and run it. Submit this exercise as .txt file*

*5 marks*

**Exercises 2.** Write a program that takes people's names and assigns them each a key consisting of a unique number. It should do this until told to stop. This information will be stored in a dictionary, and your program will end with printing the keys associated with each name.

*10 marks*

## 2 Abstract Data Types

If you are not sure what a queue or a stack are, review your lecture notes.

**Exercises 3 (Optional).** Write a function that prints out every member of a list on a different line. Do not index through the list, as in do not use `list_name[i]` at any point in your code.

**Exercises 4.** Using lists and lambda functions, write without using objects or the built in method, a priority queue datatype. *Hint: Usage could be of the form:*

```
>>>q=queue()
>>>q('add')((5,6))
>>>q('dequeue')
```

*15 marks*

**Exercises 5.** Do the same for a stack.

*15 marks*

**Exercises 6.** Now in a similar style make a bank account datatype. This datatype should provide at least three services of the type that a bank account user might need.

*20 marks*

### 3 Sorting

**Exercises 7.** Implement Selectionsort.

*10 marks*

**Exercises 8.** Implement Mergesort.

*10 marks*

**Exercises 9.** Implement the following Bozosort algorithm:

```
Take a list.
  If list is not sorted:
    switch the values in two random positions
  If list is not sorted:
    recurse
  else
    end
```

What do you think the complexity of this is? Would you ever use it? **Include the responses to those questions as comments**

*15 marks*

### 4 Optional Exercises

**Exercises 10.** Implement Bubblesort.

**Exercises 11.** Implement an **in-place** Quicksort (previous exam question).