VE477
Introduction to Algorithms

Lab 2
Manuel — UM-JI (Fall 2021)

Goals of the lab

- Efficient C implementations
- Object oriented programming in Python
- More advanced OCaml topics

1 C programming

Using the C standard library write efficient implementations of

- The `Union-Find` data structure.
- Kruskal's algorithm.
- Prim's algorithm for solving the Minimum Spanning Tree problem.

Consider the complexity of Kruskal and Prim algorithms and then compare how do they do perform in practice.

Hint: run the implementations on various types of graphs (size, sparse, dense...).

2 Getting acquainted with OCaml

For this week we want to deepen our knowledge of OCaml and be able to write our first few short programs using this new language!

2.1 More documentation

- How to define an anonymous function in OCaml? When to define and use anonymous functions?
- How should variables and functions be named (e.g. capital letters, underscore)?
- What is a module? How does it differ from an class in OOP?
- Life without arrays is not simple. How can `Lists` help?
- What are maps and iterators? How to best use them?
- Foldings are very powerful features. Explain why and provide a simple example different from the ones in the documentation.
- What is tail recursion? Why is it an important point in functional programming?
- What does `ref` mean, and when to use this keyword?
- What are *functors*, how do they relate and differ from templates in C++?
- How to define new types?
- What are sum and product types? How do they help improving coding quality?

Places of interest:

- Comparison of modules and objects

- OCaml tutorial pages: OCaml Programming Guidelines, Modules, Lists, Pointers, and Functors.

2.2 Coding!

As you probably now understand, functional languages are especially suitable for implementing recursive algorithms. In particular the `List` module can help making the code very compact.

- Implement Quick-sort in OCaml. *Hint:* `List.partition`.
- Briefly explain how your implementation works.
- How efficient is your implementation? Argue on your answer.

3 Interview problems

1. How to determine the square root of a number n, accurate to five decimal places? Do not use any packages like `numpy` or headers like `math.h`.
2. Find the longest substring without duplication. Each character should appear only once in this substring.

   Examples:

   - Input: `abcabcbb`, output: `abc` with length 3;
   - Input: `??????`, output: `?` with length 1;
   - Input: `sllktt`, output: `lkt` with length 3;