COMP 3958: Lab 6

Submit a zip file containing the `kvtree` directory & the file `primes.ml`. The `kvtree` directory should contain all source files for question 1. Do not submit `_build` directories. You may use functions from the standard OCaml library (but not from other libraries). If your program does not build, you may receive no credit for it. Be sure to comment any part of your code that may not be clear to the reviewer. Note also that you may be required to explain your work. Maximum score: 13

1. The implementation of binary search trees with keys & values (that we'll call key-value trees) in lab 4 uses a comparsion function. In this exercise, you are asked to implement a module named `Kvtree` with a functor named `Make` to create modules of key-value trees. `Make` takes a module of type `OrderedType` (the type of the key) which is defined as follows:

   ```
   module type OrderedType = sig
     type t
     val compare : t -> t -> int
   end
   ```

   Make the key-value tree type abtract.

   You'll need to implement the 4 functions specified in lab 4, suitably modified:

   - they should be renamed to `insert`, `find`, `delete` & `of_list`
   - `insert`, `find` & `delete` take a tree as an argument; make that tree the last argument
   - they do not use labelled arguments

   You will also need to provide 3 additional functions:

   (a) `empty` that returns an empty tree

   (b) `is_empty` that tests whether a tree is empty

   (c) `to_list` that returns a list of key-value pairs that represents the tree in ascending order of keys as specified by the `compare` function for keys

   Create a directory named `kvtree` and put your implementations in that directory. There should be a file named `kvtree.ml` and its corresponding interface file `kvtree.mli`. You may use additional files if necessary. Make sure your system can be built using the command: `corebuild kvtree.cmo`.

2. For this part, you are asked to implement a stand-alone program that uses the OCaml `Map` module (with the `Make` functor) in the standard library to find the size of the largest set of 6-digit primes that are permutations of one another. This is basically question 2 of the previous lab. However, you do not need to find the 6-digit primes. A list of all 6-digits primes is provided in the file `6-digit-primes.txt`. Your program needs to read them in (via `stdin` using I/O redirection) & process them.

   Name your file `primes.ml` & make sure it can be built using the command: `ocamlbuild primes.native`.

   See documentation of the `Map` module at

   `https://ocaml.org/api/Map.html`