Programming languages (TC-2006)

Midterm Exam - Haskell

Date: April 20, 2021

This exam contains four categories of problems. In the cases where a category contains more than one problem, you can choose which problem to solve. When you submit your solution, the selected problem per category must be implemented using the template distributed along with this document. Please be warned that failing to provide the output as requested will invalidate your solution. Also, do not remove the legend "Not yet implemented" unless you plan to solve the corresponding problem. In other words, it must be clear which problem you have chosen to solve. In case two problems are chosen in the same category, the points for that category will be canceled.

In all the cases, assume that the input is always in the proper format, so no additional validations are requested on the input. To be sure that your solution will get all the points, try the test cases contained in the template distributed along with this document. Please note that these test cases will be used for grading. Then, you are not requested to cover extreme cases, only the ones presented in the template. However, be warned that changing the names of the functions or their number of arguments, or failing to provide the output in the expected format described in the template will invalidate your solution.

1 Category I (20%)

1.1 `minusIndex`

Code a function in Haskell that receives a list of integers and returns a list of those numbers but decreased by its position in the original list. Please consider that the first element of the list is located at position 0.

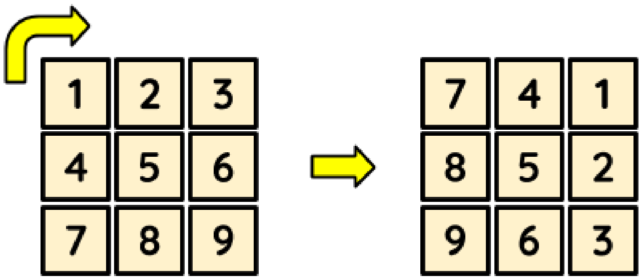
1.2 `evenList`

Write a function in Haskell that receives a list of integers and returns a list of pairs. The pairs contain the corresponding number in the list and a boolean value that indicates if such a number is even or not.

2 Category II (25%)

2.1 `rotate`

Provide a function in Haskell that 'rotates' a matrix of integers, 90 degrees to the right, as described in the following picture:

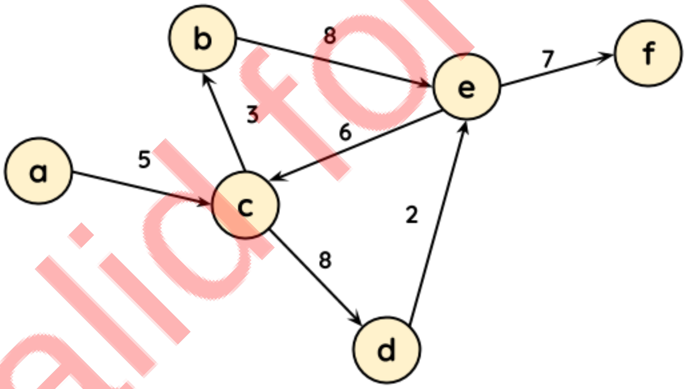


$$\begin{bmatrix}1&2&3\\4&5&6\\7&8&9\end{bmatrix} \rightarrow \begin{bmatrix}7&4&1\\8&5&2\\9&6&3\end{bmatrix}$$

For this problem, use a 'by-row' representation for the matrices.

2.2 `distance`

In this problem you will work with directed weighted graphs which are represented by using nodes and vertices. For example, the graph:



can be represented in Haskell as:

```
[
    ('a', 'c', 5), ('b', 'e', 8), ('c', 'b', 3),
    ('c', 'd', 8), ('d', 'e', 2), ('e', 'c', 6),
    ('e', 'f', 7)
]
```

Write a function in Haskell that calculates the distance of a path provided as argument (the sum of the weight of the edges involved). For example, one possible path from `a` to `f` will be represented as `[('a', 'c'), ('c', 'b'), ('b', 'e'), ('e', 'f')]`. The distance of such a path is 23.

3 Category III (25%)

3.1 pass

Imagine you are given a four-column table with information about a class. In this table, the rows contain information about the student ID and the grades obtained in three midterm exams. For example, the table:

ID	G1	G2	G3
796623	62	84	91
782544	81	89	77
790256	59	62	63
799610	75	68	82
791313	85	56	92
786621	98	97	92
809706	91	72	80

The rows in this table are represented by the following user-defined data type:

```haskell
data Record = Record Int Double Double Double deriving Show
```

Then, the table can be coded in Haskell as:

```haskell
[
   (Record 796623 62 84 91), (Record 782544 81 89 77),
   (Record 790256 59 62 63), (Record 799610 75 68 82),
   (Record 791313 85 56 92), (Record 786621 98 97 92),
   (Record 809706 91 72 80)
]
```

Write a predicate in Haskell that receives a table represented as in the previous example and indicates if a particular student has passed the course or not. The passing grade for the course is 70, and the final grade is calculated as the average grade of the three midterm exams.

3.2 tsort

Imagine you are given a two-column table with information about a class. In this table, the rows contain information about the student ID and the grade obtained in such a course, respectively. For example, the table:

ID	Grade
796623	92
782544	83
790256	62
799610	75
791313	85
786621	98
809706	94

The rows in this table are represented by the following user-defined data type:

```
data Grade = Grade Int Int deriving Show
```

Then, the table can be coded in Haskell as:

```
[
   (Grade 796623 92), (Grade 782544 83),
   (Grade 790256 62), (Grade 799610 75),
   (Grade 791313 85), (Grade 786621 98),
   (Grade 809706 94)
]
```

Write a function in Haskell that receives a table represented as in the previous example and returns the table sorted by the grade (in ascending order). Feel free to use the sorting algorithm of your preference.

4 Category IV (30%)

Write a function in Haskell that mimics the behavior of `apply`, from Racket (or `foldl1`, from Haskell). The function receives a two-argument function and a list, and the result is the application of the function to all the elements in the list (in case of doubts about the behavior of `foldl1`, revise the course materials).

Deliverables

Prepare an HS file that contains the functions requested and submit it to Canvas. **Please, do not submit other formats but HS**. To prepare your HS file, use the code template distributed along with this document. The template contains some test cases for each function to help you verify that your codes work as requested.

I promise to apply my knowledge, strive for its development, and not use unauthorized or illegal means to complete this activity, following the Tecnológico de Monterrey Student Code of Honor.