E-402-STFO PROBLEMS FOR MODULE C

CREATED BY HENNING ULFARSSON

This is the second module concerned with dynamic programming. In some problems I also give you a hint towards a solution that perhaps avoids using dynamic programming.

You get a perfect score for this module by getting 30 points or more.

Please note: You are are now allowed to use the built-in function `Partitions(n)` to solve the problems in this section.

Problem 1 (10 points)**.** Write a function `mCp1(M)` that solves Project Euler problem 82 (`http://projecteuler.net/problem=82`). The input `M` should be an $n \times n$ matrix containing positve integers.

```
Input: mCp1(matrix[[1,1,9], [9,1,1], [9,9,1]])
Output: 4
```

Here you can use dynamic programming or think about this problem in terms of a directed graph with weights on the edges. You are trying to find the path from the "vertices in the first column" to the "vertices in the last column" with the smallest total weight.

Problem 2 (10 points)**.** Write a function `mCp2(n)` that solves Project Euler problem 85 (`http://projecteuler.net/problem=85`). The input `n` should be a positive integer. The returned value is the area of the smallest rectangular grid containing the number of rectangles that best approximates `n`.

```
Input: mCp2(18)
Output: 6
```

Here you can use dynamic programming or think about how many choices you have when placing a small rectangle in a large rectangle. The smaller rectangle is determined by choosing two vertical lines and two horizontal lines.

Problem 3 (15 points)**.** Write a function `mCp3(n)` that solves Project Euler problem 178 (`http://projecteuler.net/problem=178`). The input `n` should be a positive integer. The returned value is the number of pandigital step numbers less than `n`.

```
Input: mCp3(10**10)
Output: 1
```

Sorry, I only know how to do this one with dynamic programming! Note that all testcases will input a power of 10.

Problem 4 (20 points)**.** Write a function `mCp4(n,L)` that solves Project Euler problem 329 (`http://projecteuler.net/problem=329`). The input `n` should be a positive integer, determining the array `[1,2,...,n]` that the frog can jump around on. The input `L` is a (non-empty) list of `0`'s and `1`'s, determining the croaks Susan is hoping to hear, `1` denotes a prime and `0` denotes a non-prime. The Project-Euler

Date: Updated September 26, 2013.

testcase corresponds to `n = 500` and `L = [1,1,1,1,0,0,1,1,1,0,1,1,0,1,0]`. The returned value is the probability p/q of Susan hearing the sequence of croaks given by `L`. Use the built-in type `sage.rings.rational.Rational` which is the default when you divide two integers.

```
Input: mCp4(2,[1,0])
Output: 5/18
```

It is very tempting to simulate this problem and try to guess `p` and `q`! Therefore testcases will pass if your output is close enough to the correct value.

School of Computer Science, Reykjavik University, Menntavegi 1, 101 Reykjavík, Iceland

E-mail address: `henningu@ru.is`