

COMP 102A 2014, Assignment 3

Due Wednesday November 12th 2014

[30%] 1. We know that the Post Correspondence Problem cannot be solved in general by any algorithm. Here is a few facts about variations on PCP:

* PCP is decidable if we restrict the strings to come from an alphabet with a single symbol (ex: $\Sigma = \{a\}$),

* PCP is decidable if we restrict the tile set to contain only two tiles (ex: $(ab/aaa), (aa/bb)$).

Show that if we restrict PCP according to both of these constraints, "the strings come from an alphabet with a single symbol" and "the tile set contains only two tiles", then it is decidable (ex: $(aa/aaa), (aa/aa)$). Give an explicit algorithm that decides whether a two-tile input $(u_1/v_1), (u_2/v_2)$, where u_1, v_1, u_2, v_2 are strings of a 's, form a positive or negative instance of PCP. ($|s|$ tells you the number of symbols of a string as $s.length$ in JavaScript.)

JAVAscript

[30%] 2. Write a JAVAscript function **sort** that inputs a string **w** and outputs another string **z** such that if you consider the characters of **z** they are the same as those of **w**, but in alphabetical order.

Example:

`sort("aviation")` should output the string "aaiinotv".

Your function should work for arbitrarily long strings. We don't care what happens if the input is not of the right format. A few examples will be made available to you. Your function should be executed on these inputs and return the correct outputs.

[40%] 3. Write two JavaScript functions **Eotp** (resp. **Dotp**) that inputs two strings **m,k** (resp. **c,k**) and outputs another string **c** (resp. **m**) such that if you consider the characters of **m,k** and **c**, we have the relation $c[i] = m[i] + k[i]$ where the sum is interpreted that "A" corresponds to 0, "B" corresponds to 1, ..., "Z" corresponds to 25, and addition is modulo 26. All of **c,k,m** are strings made of capital letters only. **Dotp** is the inverse of **Eotp** computing $m[i] = c[i] - k[i]$.

Example:

```
Eotp(  
  "VERNAMINVENTEDTHEONETIMEPAD",  
  "ABCDEFGHIJKLMNOPQRSTUVWXYZ")
```

should output the string

```
c="VFTQEROUDNXEQQHWUFFXNDIBNZD"
```

which is the One-time-pad sum of the value **m** encrypted by key **k**. Your functions should work for arbitrarily long strings. We don't care what happens if the inputs are not of the right format. A few examples will be made available to you. Your functions should be executed on these inputs and return the correct outputs.