



Walter Cazzola

[Home Page](#)
[ADAPT Lab.](#)
[Curriculum Vitae](#)
[Research Topic](#)

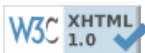
Didactics

Publications

Funded Projects

Research Projects

Related Events



Exam of Advance in Programming

27 February 2013

Disclaimer. Note that to have a running solution for an exercise is not enough: you need a well-cooked solution that proves your ability to use what explained during the classes. The worth of the 2 exercises is the same (15 points). To pass the exam you have to do both exercises. The submissions with only one exercise will not be evaluated at all.

Exercise 1: Ruzzle's Solver.

Ruzzle is a grid (4×4) based game where each element of the grid is a letter and the aim of the game consists in determining all the possible English words out of all possible paths in the grid (where a path is a sequence of adjacent grid elements).

The exercise consists of writing a function `ruzzles` that calculates all the English words hidden in a given grid with the following simplifications/constraints:

- the minimum length for a word is 3 characters;
- an entry is considered adjacent to another if it is immediately at left, at right, above or below it (no diagonals are considered); note that a word can also be hidden in the grid in its reversed form (i.e., it can be read from right to left);
- no entry can be considered twice in the same word;
- the words should be checked against this [dictionary](#);
- the list of resulting words must be alphabetically sorted;

The input is a list of 4-character strings where each string represents a row in the grid and their position in the list mimic the position in the grid.

The following is an example of the expected computation.

```

from ruzzle import *

if __name__ == "__main__":
    print(ruzzles(["walk", "moon", "hate", "rope"]))
    print(ruzzles(["abbr", "evia", "tion", "alba"]))
    print(ruzzles(["abse", "imtn", "nded", "ssen"]))
    print(ruzzles(["reca", "rwar", "aazp", "syon"]))
    print(ruzzles(["abst", "oime", "uesl", "snsp"]))
    print(ruzzles(["essx", "ndet", "sigh", "raen"]))

```

```

[17:32]cazzola@surtur:~/pa/es1>python3 main-ruzzle.py
['alone', 'ate', 'atone', 'eta', 'hat', 'hate', 'knee', 'knot', 'law',
'alone', 'loom', 'lot', 'moat', 'moo', 'moon', 'moot', 'nee', 'net', 'not',
'note', 'oat', 'oaten', 'one', 'opt', 'pee', 'peen', 'rope', 'tee', 'ter',
'ton', 'tone', 'too', 'walk']
['ablate', 'ablation', 'ablativ', 'alb', 'alit', 'alive', 'ate', 'ban',
'bio', 'boil', 'bra', 'bran', 'eta', 'evil', 'ion', 'late', 'lion', 'lit',
'live', 'live', 'nab', 'naive', 'oblate', 'oil', 'ran', 'tali', 'tea',
'vet', 'via', 'vita', 'vital']
['abs', 'absent', 'absentee', 'absentminded', 'absentmindedness', 'aim',
'end', 'mind', 'minded', 'nee', 'need', 'needs', 'nest', 'see', 'seed',
'send', 'sent', 'tee', 'teen']
['ace', 'away', 'awe', 'car', 'caraway', 'carp', 'caw', 'err', 'race', 'r',
'raceway', 'raw', 'rec', 'war', 'warp', 'way', 'ways', 'yaw', 'yon']
['abs', 'abstemious', 'abstemiousness', 'bio', 'else', 'let', 'met', 'mie',
'stem', 'sue']
['aegis', 'aid', 'aide', 'dig', 'dis', 'ear', 'end', 'gear', 'get', 'ides',
'near', 'nearsighted', 'nearsightedness', 'raid', 'send', 'set', 'side',
'sigh', 'sight', 'sighted']

```

Exercise 2: Testing the Ruzzle's Solver.

Given the specification of Exercise 1 in this exercise you have to test the correctness of your solution (i.e., the result of the `ruzzles` function).

Hint, some of the properties that can be tested are:

- the resulting list of words is alphabetically sorted;
- the words in the result are really English words, i.e., they are in the given dictionary;
- no word in the result are shorter than 3 characters and longer than the maximum number of

characters available

- each word in the result is really hidden in the grid (use a different algorithm than the one used in exercise 1, otherwise I will consider the exercise as wrong)
- no grid element is used twice (or as an approximation: there should be as many occurrence of an element in the word as in the grid or less).

Please submit a complete program including the main to execute the tests.

Last Modified: Sat, 30 Mar 2013 22:12:15

ADAPT Lab. 