Algorithmics
Midterm #2 (C2)

Undergraduate 1^{st} year (S2)
Epita

22 February 2017 - 9 : 30

Instructions (read it) :

□ You must answer on **the answer sheets provided.**

- No other sheet will be picked up. Keep your rough drafts.
- Answer within the provided space. **Answers outside will not be marked**: Use your drafts!
- Do not separate the sheets unless they can be re-stapled before handing in.
- Penciled answers will not be marked.

□ The presentation is negatively marked, which means that you are marked out of 20 points and the presentation points (maximum of 2) are taken off this grade.

□ **Code:**

- All code must be written in the language `Python` (no C, Caml, Algo or anything else).
- **Any `Python` code not indented will not be marked.**
- All that you need (functions, methods) is indicated in the **appendix** (last page)!

□ Duration : 2h

CALVIN, I DON'T WANT YOU UP IN THAT TREE.
WHY NOT?
SOME OF THE BRANCHES ARE DEAD AND THEY MIGHT BREAK. COME ON DOWN.
MOM SPOILS EVERYTHING.
THIS ISN'T QUITE THE SAME, IS IT?
©1986 Universal Press Syndicate
WATTERSON
7-12

Lecture

Exercise 1 (Be daring... - *3 points*)

Let the following binary tree, represented (statically) in hierarchical form, be:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
T	E	C	D	E	F	R	A		R	S		A	E	E		U				E	

1. Draw the corresponding tree.
2. Considere a depth-first traversal of this tree and give the inorder node list.
3. Give the occurrence representation ($B = \{\varepsilon, 0, 1, 01, 10, \ldots\}$) of this tree.

Matrices

Exercise 2 (Maximum Gap – *5 points*)

In this exercise, the *gap* of a list is defined as the maximum difference between two values of the list. For instance, in the matrice below, the *gap* of the first line is 13.

Write the function that returns the maximum gap of the lines of a matrice (assumed non empty).

Example of result with the matrice `Mat1` *on the right:*

```
    >>> maxGapMatrix(Mat1)
    19
```

Indeed the maximum gap is the one of the middle line (19 = 14 - (-5)).

1	10	3	0	-3	2	8
-1	0	1	8	5	0	-4
10	9	14	1	4	-5	1
10	-3	7	11	6	3	0
7	8	-5	1	5	4	10

`Mat1`

Exercise 3 (Synergistic Dungeon – *4 points*)

The demons have captured the knight **(K)**. He is being held prisoner in the bottom-right corner of a dungeon consisting of $m \times n$ rooms laid out in a 2D grid. Our valiant princess **(P)** enters the donjon by the top-left room, she must fight his way through the dungeon to rescue the knight.

The princess has an initial number of health points that is given by a positive integer. If at any point that number drop to 0 or below, she dies immediately.

Some of the rooms are guarded by demons (positive integers), so the princess loses health points upon entering these rooms; other rooms are empty (0's).

In order to reach the knight as quickly as possible, the princess decides to move only rightward or downward at each step.

For example, given the dungeon below, the initial number of health points the princess must have is at least 7, assuming she follows the optimal path *right* → *right* → *down* → *down*.

(P) 2	3	0
2	10	0
3	0	1 **(K)**

Write the function `dungeon`(M), with M the matrix that represents the dungeon (not empty), that determines the minimum initial number of health points the princess must have to rescue the knight (if she chooses the optimum path).

Add-on/clue: you do not need to build another matrix, the one in parameter can be modified.

Binary Trees

Exercise 4 (Tests – *8 points*)

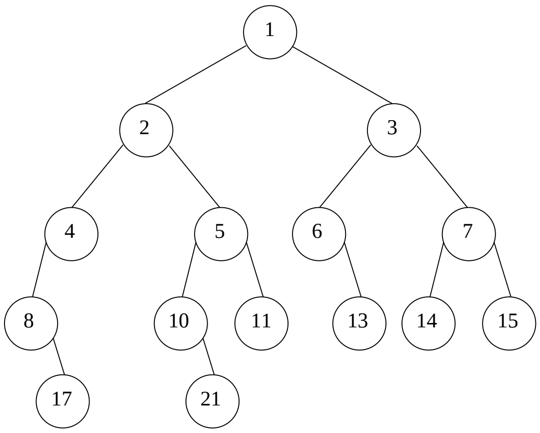

Figure 1: Binary Tree B

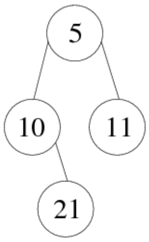

Figure 2: A sub-tree of B

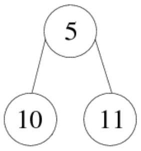

Figure 3: Not a sub-tree of B

1. Write the function `equal` that tests whether two binary trees are identical. That is, if they contain same values in same nodes.
2. Write the function `isSubTree(`S`, `B`)` that tests whether the binary tree S is a subtree of the binary tree B. Assume there are no two identical keys in B.

Appendix

Binary Trees

The binary trees we work on are the same as the ones in tutorials.

- `None` is the empty tree.
- The non-empty tree has 3 attributes: `key`, `left`, `right`.

Authorised functions and methods

On lists:

- `len`
- `append`

Others:

- `range`
- `abs`
- `min` and `max`, but only with two integer values!

Your functions

You can write your own functions as long as they are documented (we have to known what they do).

In any case, the last written function should be the one which answers the question.