

Kai-Feng Chen
Po-Ya Yang
National Taiwan University

PROGRAMMING & NUMERICAL ANALYSIS

Quizzes

2016

ALL YOU NEED TO KNOW

- There are 5 short quizzes in the following slides. You have to solve at least **4 out of 5**.
You are strongly recommended to solve all 5 problems ^_^.
- The deadline for uploading is **13:00, April 28**. You have 2 full weeks to solve them. **Late Assignments are NOT accepted.**
- Please download the template code from CEIBA and upload your answers to the system (at the homework area).
- If you do not understand any of the quizzes or anything associated with the rules, please contact the TA or me.

QUIZ I:

Palindromic Numbers

- A palindromic number is a number which reads the same backward and forward. For examples, 363 and 68986. Start from 11, 22, 33, 44... what's the 100th, and 1000th palindromic number?

QUIZ 2:

Parallelogram Words

- Write a function that takes an arbitrary string and print the result in parallelogram manner as the example below,

```
>>>parallelogram_word( 'TEST' )  
  T  
 TE  
TES  
TEST  
EST  
ST  
T
```


QUIZ 3:

Decoding

- We have the following code mapping decryption key,

A	B	C	D	E	F	G	H	I	J	K	L	M

N	O	P	Q	R	S	T	U	V	W	X	Y	Z

(letter above equals below, and vice versa)

Let capital letters be translated to capital ones, while lower case letters be translated to lower case ones. And the symbols will remain the same. One can translate a simple sentence like “**Hello, World**” to be:

Uryyb, Jbeyq

QUIZ 3:

Decoding (cont.)

- With the same translation rule, what is the real content of the following codes? Print it out on the screen.

```
V jnf greevoyr va Ratyvfu. V pbhyq abg
fgnaq gur fhowrpg. Vg frrzrq gb zr
evqvphybhf gb jbeel nobhg jurgure lbh
fcryyrq fbzrguvat jebat be abg, orpnhfr
Ratyvfu fcryyvat vf whfg n uhzna
pbairagvba - vg unf abguvat gb qb jvgu
nalguvat erny, nalguvat sebz angher.
Evpuneq C. Srlazna
```


QUIZ 4:

Find the best traveling path

- There are 7 cities as shown on the map. Some of them have standard routes connected, but some of them do not. Implement a function, which takes **two strings** (city names) as the argument, and print out the shortest route between them.
- For example, from “ANNUMINAS” to “BREE”, your answer should be shown as:

The shortest path: ANNUMINAS > 12 km > FORNOST > 11 km > BREE

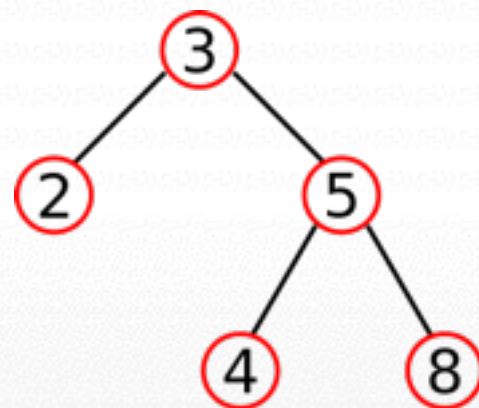


QUIZ 5:

Tree Builder

- In this problem, you are going to build a `tree_builder` function that generates tree structure in the **string format** from a **list of all-different positive integers**.

Tree consists of **nodes**. Each node contains a value. And some of the nodes have branches connecting to other nodes.



The figure above is a tree built from the list `[3, 5, 4, 2, 8]`.

QUIZ 5:

Tree Builder - Building Rules

- A. The first number in the list is the “root” node.
- B. One node can have at most two branches. The number in the right-branch node is larger than the number in the mother node, The number in the left-branch node is smaller than the number in the mother node.
- C. When building one number on an existing tree, start from the “root” node. Compare the number you want with the number in the node, and then move the number you want to the correct branch decided with rule B. (larger to the right, and smaller to the left) Finally, create a new node for the number if available.

QUIZ 5:

Tree Builder - Building Example

Take the building process from the list [3, 5, 4, 2, 8] for example,

- a. At first, we have (3) as our “root” node.
- b. When building (5), we create a new node on the right branch of the “root” node.
- c. When building (4), we first move it to the right branch of the “root” node, (5). And create a new node on the left branch of the (5) node.
- d. When building (2), the left branch of the “root” node is now still available. We create a new node on the left branch of the “root” node.
- e. When building (8), we first move it to the right branch of the “root” node, (5). And create a new node on the right branch of the (5) node

③

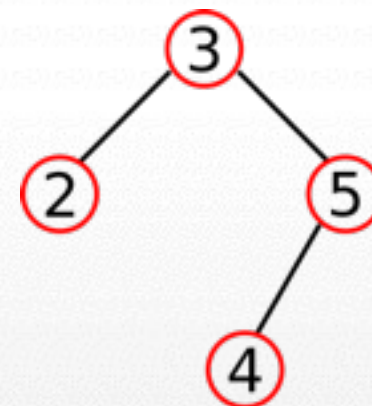
a.



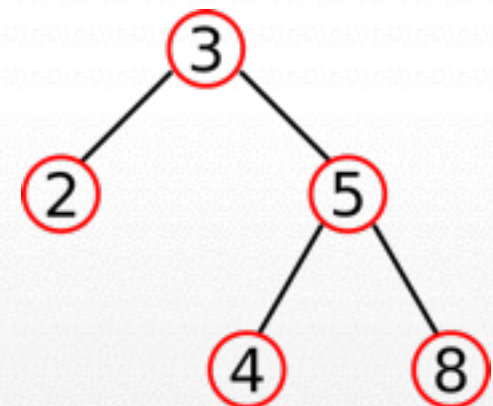
b.



c.



d.



e.

QUIZ 5:

Tree Builder - String Representation

In this problem, we try to use **string** to record the tree process described above. Here is the format,

- A. Each node is written as (left,value,right).
- B. Branches left and right could be empty, which is specified with dash -.
- C. If branch is not empty, it should contain another node.

QUIZ 5:

Tree Builder - Building Example with String

Let's review the building example above in string representation,

a. At first, we have (3) as our “root” node.

```
'(-,3,-)'
```

b. When building (5), we create a new node on the right branch of the “root” node.

```
'(-,3,(-,5,-))'
```

③



a.

b.

QUIZ 5:

Tree Builder - Building Example with String (cont.)

- c. When building (4), we first move it to the right branch of the “root” node, (5). And create a new node on the left branch of the (5) node.

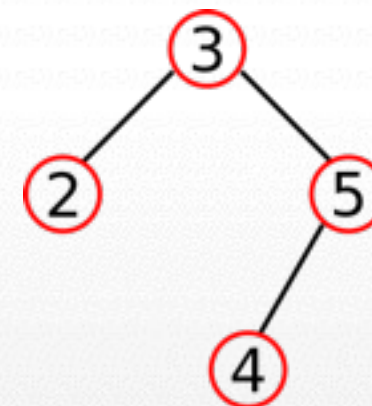
```
' (-, 3, ((-, 4, -), 5, -)) '
```

- d. When building (2), the left branch of the “root” node is now still available. We create a new node on the left branch of the “root” node.

```
' ((-, 2, -), 3, ((-, 4, -), 5, -)) '
```



c.



d.

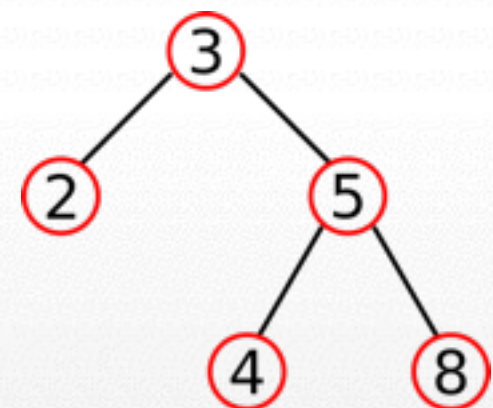
QUIZ 5:

Tree Builder - Building Example with String (cont.)

e. When building (8), we first move it to the right branch of the “root” node, (5). And create a new node on the right branch of the (5) node

```
' ( (-, 2, -), 3, ( (-, 4, -), 5, (-, 8, -) ) ) '
```

And this string is the expected answer returned by your tree_builder([3, 5, 4, 2, 8])



e.

QUIZ 5:

Tree Builder - Check Point

- In this problem, you are going to build a `tree_builder` function that generates tree structure in the **string format** from a **list of all-different positive integers**.
- Before submitting your code, please make sure it can:
 - a. Deal with an arbitrary length of the input list.
 - b. Handle positive integers with more than 1 digits.
- You can take the example below to test your code.

```
>>>tree_builder([314, 159, 265, 35, 89, 79, 323, 84, 62])  
' (((-,35,((-,62,-),79,(-,84,-)),89,-)),159,(-,265,-)),314,(-,323,-)) '
```