Oregon State University
School of Electrical Engineering and Computer Science

CS 261 – Recitation 7

Spring 2015

Outline

- AVL trees
- Assignment 5 – Heap Implementation of a ToDo List
  - File handling and standard I/O in C

Parts of this lecture were taken from:
www.cs.txstate.edu/~rp44/cs3358_089/Lectures/bst.ppt

www.cs.sjsu.edu/~lee/cs146/Asami-avl-presentation.ppt

An AVL Tree is…

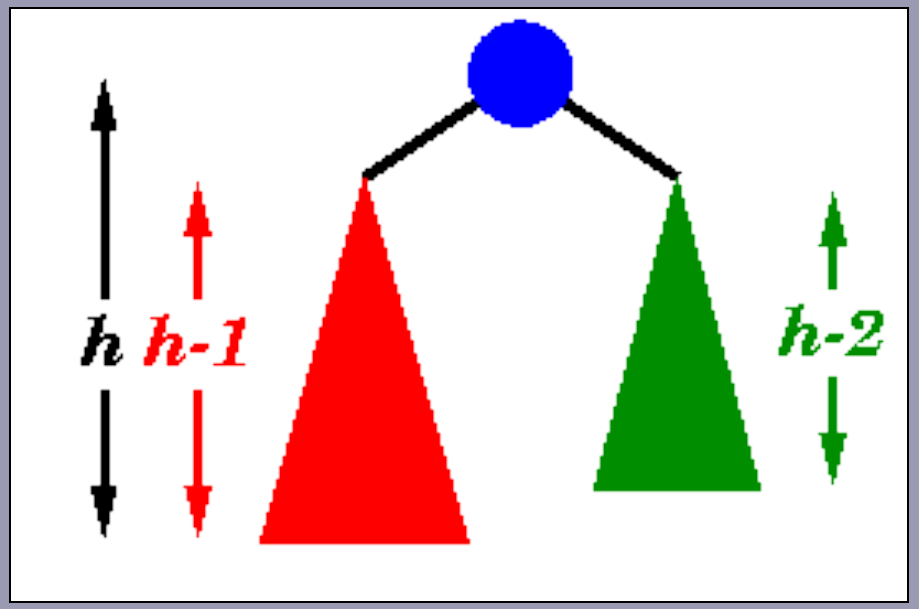
- A binary search tree with a **balance condition**.
  - The sub-trees of each node differ by at most 1 in their height
- Named after its creators (**A**delson-**V**elskii and **L**andis)

Definition of a balanced tree

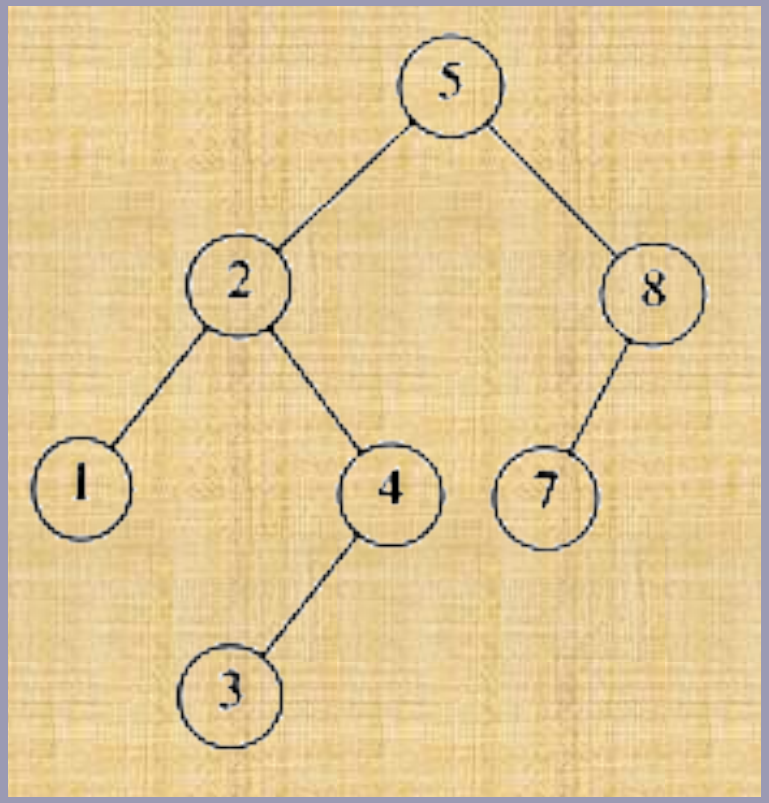
- Every node must have left & right sub-trees of heights that differ by no more than 1
  - Ensures the tree's height = log N
  - Takes O(log N) time for searching, insertion, and deletion

An AVL tree has the following properties

1. Sub-trees of each node can differ by at most 1 in their height
2. Every sub-tree is an AVL tree

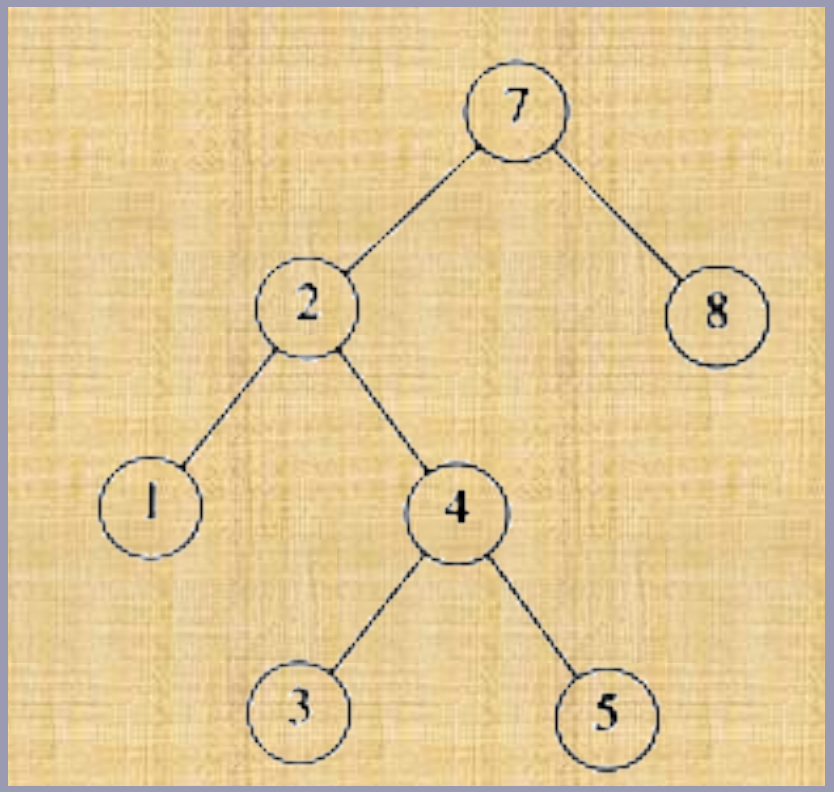


Are these AVL trees?



YES

Each left sub-tree has height 1 greater than each right sub-tree



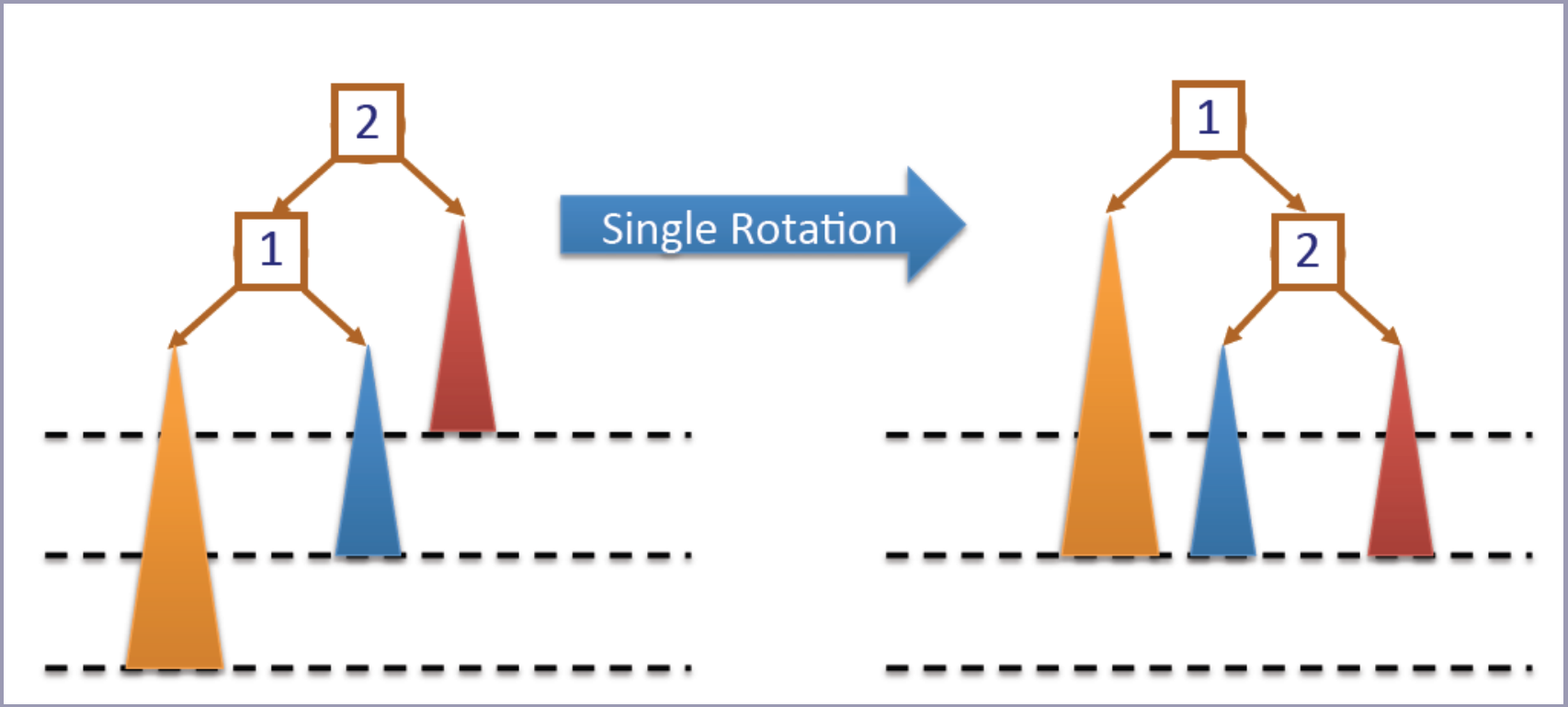
NO

Left sub-tree of the root has height 2, but right sub-tree has height 0

Insertion and Deletions

- Performed as in binary search trees
- If the balance is destroyed, rotations are performed to correct balance
  - For insertions, one single or double rotation is required.
  - For deletions, at most O(log n) rotations are needed

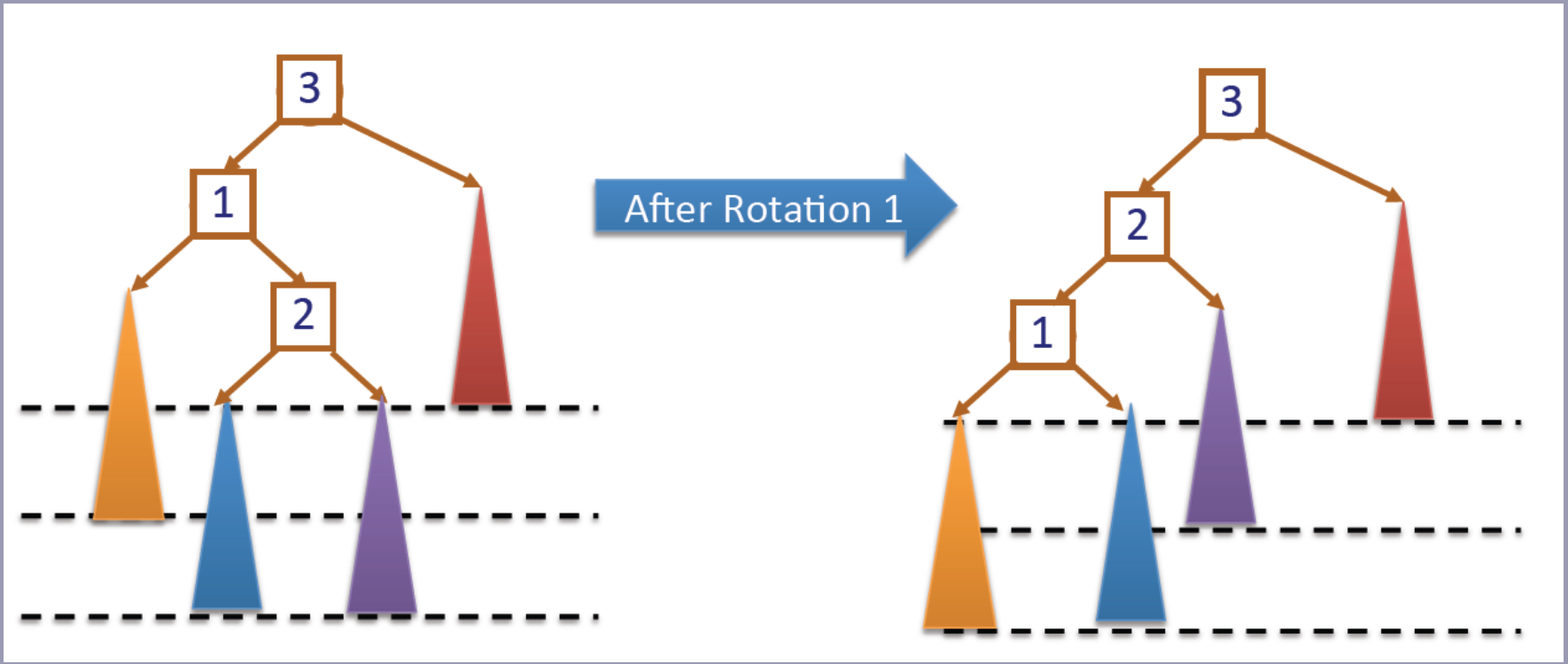
Single Rotation



Node (2) is heavy on the left child, which is also heavy on the left.

Rotate the heavy node (1) to the right.

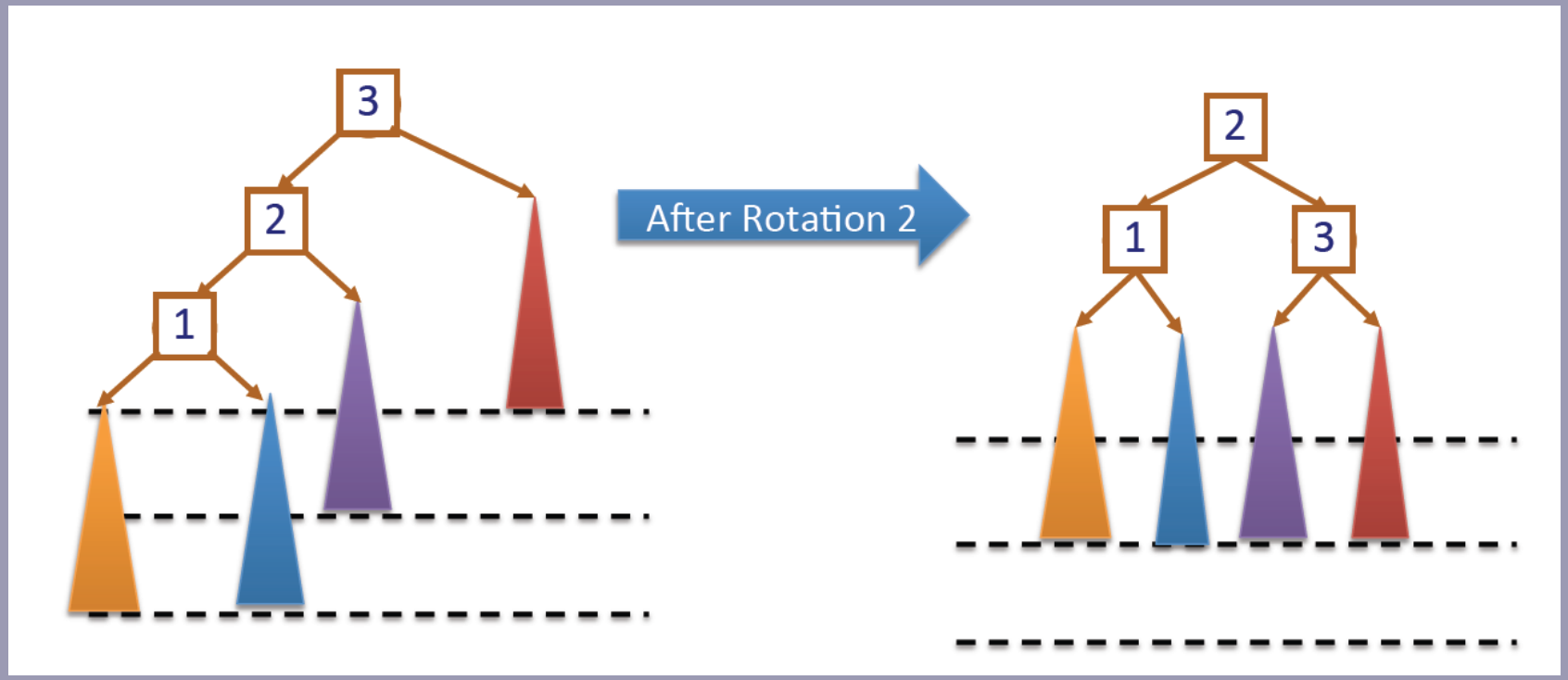
Double Rotation



Node (3) is heavy on the left child, which is heavy on the right.

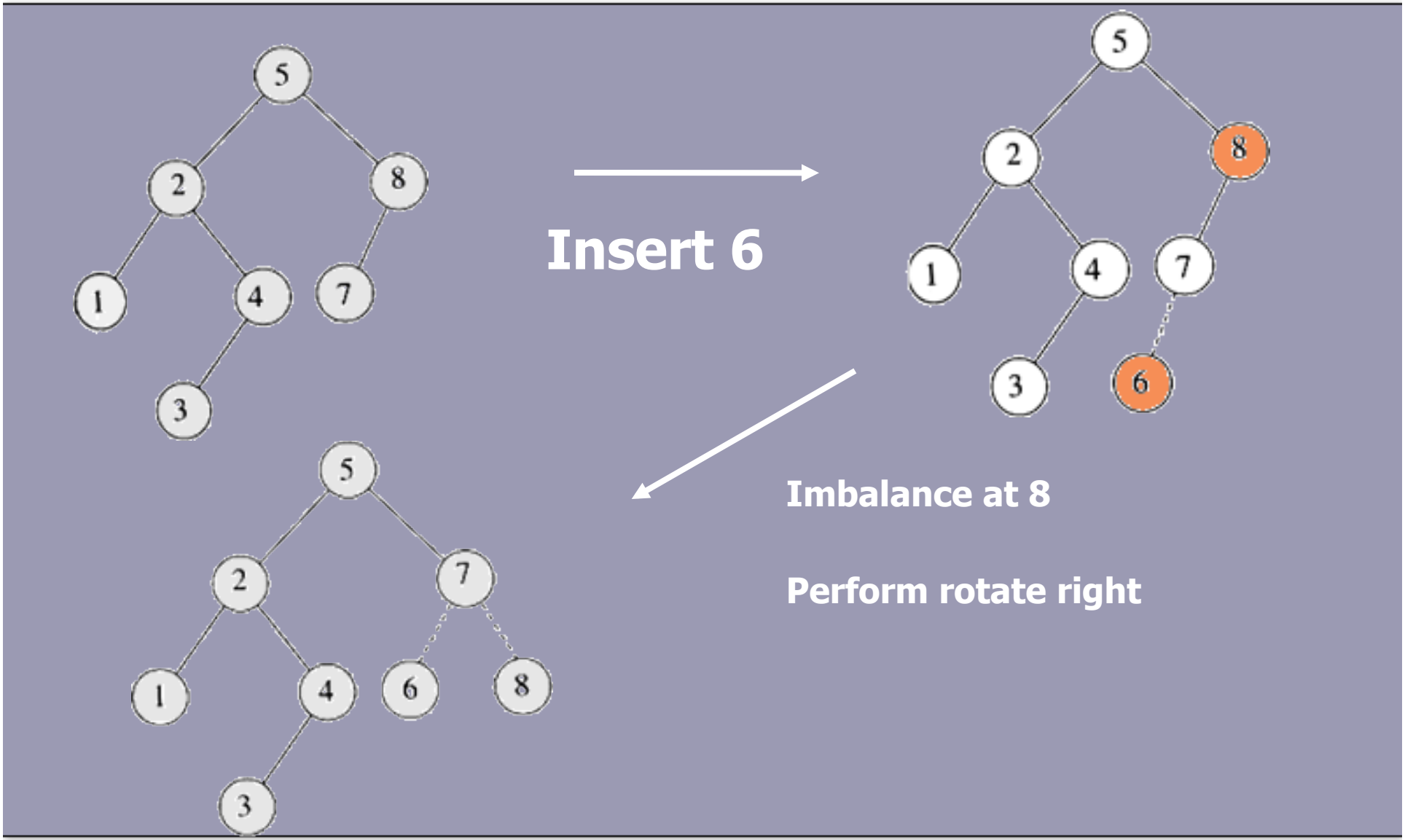
First, rotate heavy node (1) to the left

Double Rotation

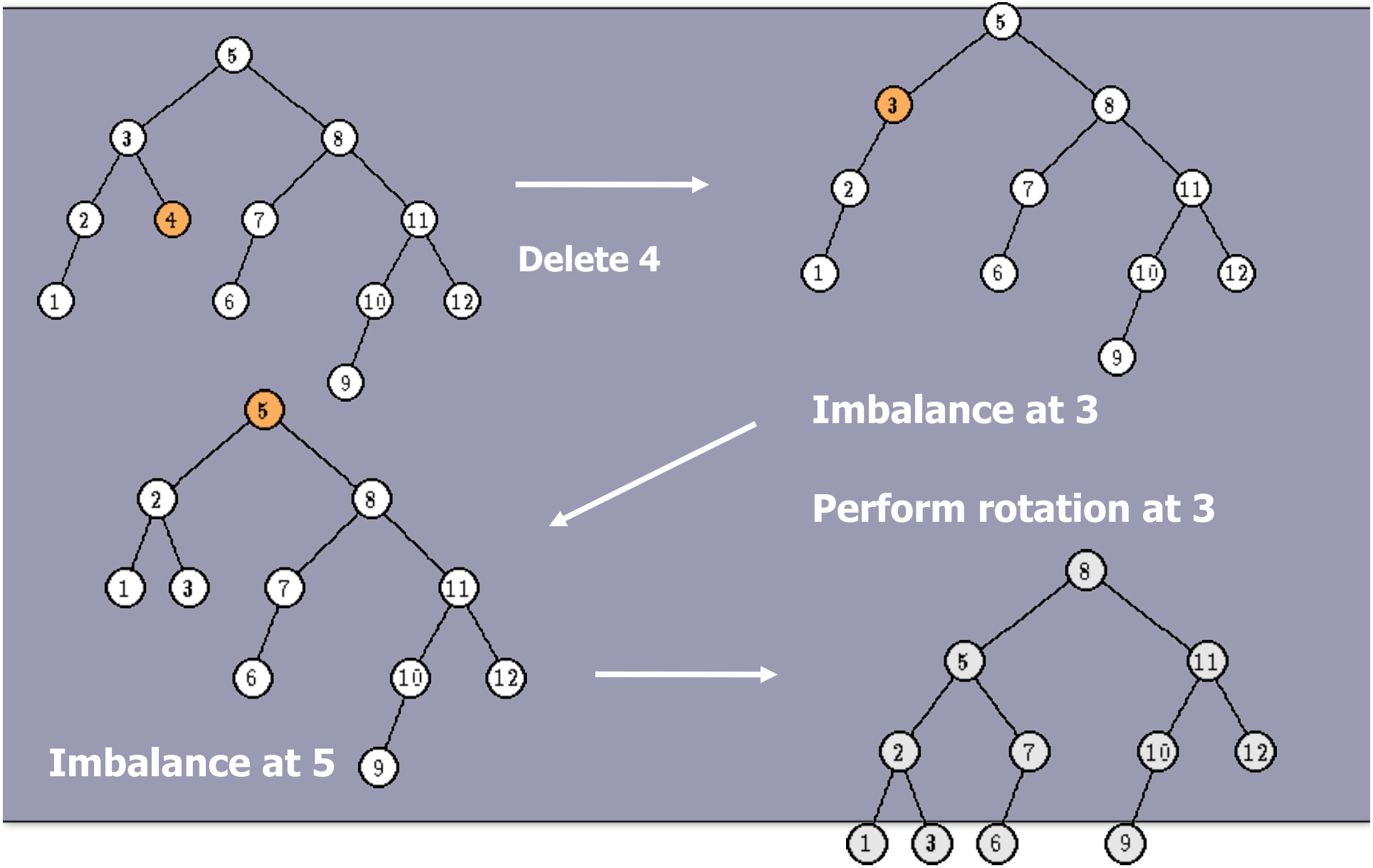


Next, rotate unbalanced node (3) to the right

Insertion


5
2
8
1
4
7
3
Insert 6
5
2
8
1
4
7
3
6
Imbalance at 8
Perform rotate right
5
2
7
1
4
6
8
3

Deletion

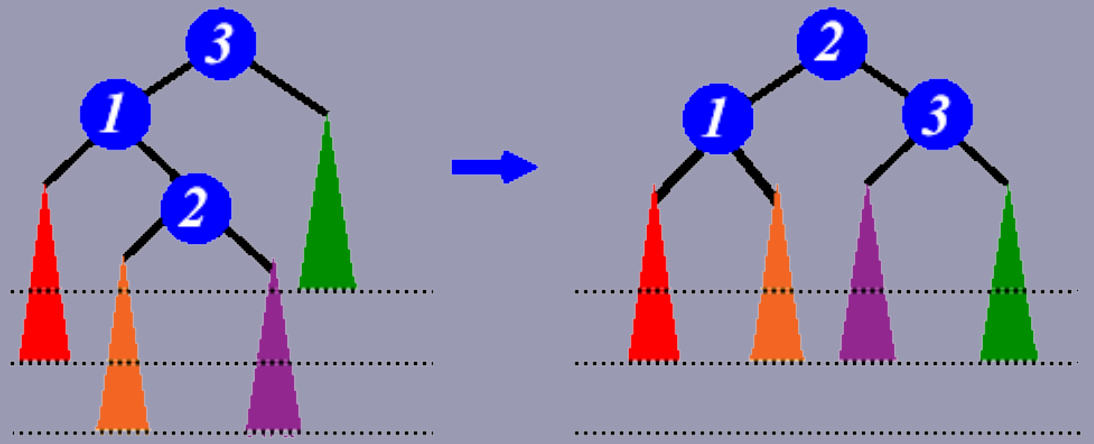


Key Points

- AVL tree remains balanced by applying rotations, therefore it guarantees O(log N) search time in a dynamic environment
- Tree can be re-balanced in at most O(log N) time

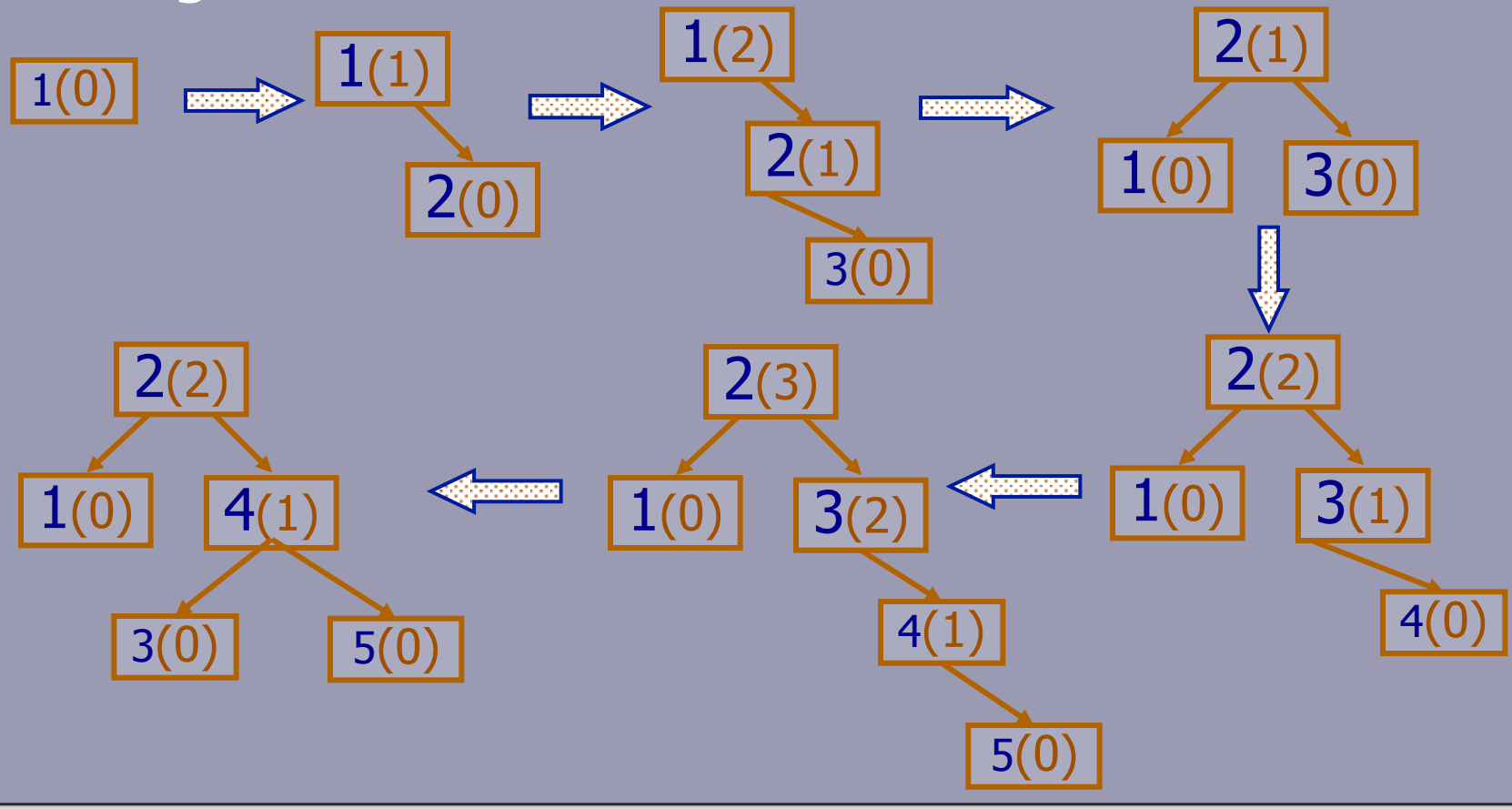
When to do a double rotation?

- Balance factor = height(left tree) – height(right tree)
- For node N, a double rotation is needed when:
  - N's balance factor is positive and N's left subtree's balance factor is negative
  - N's balance factor is negative and N's right subtree's BF is positive.

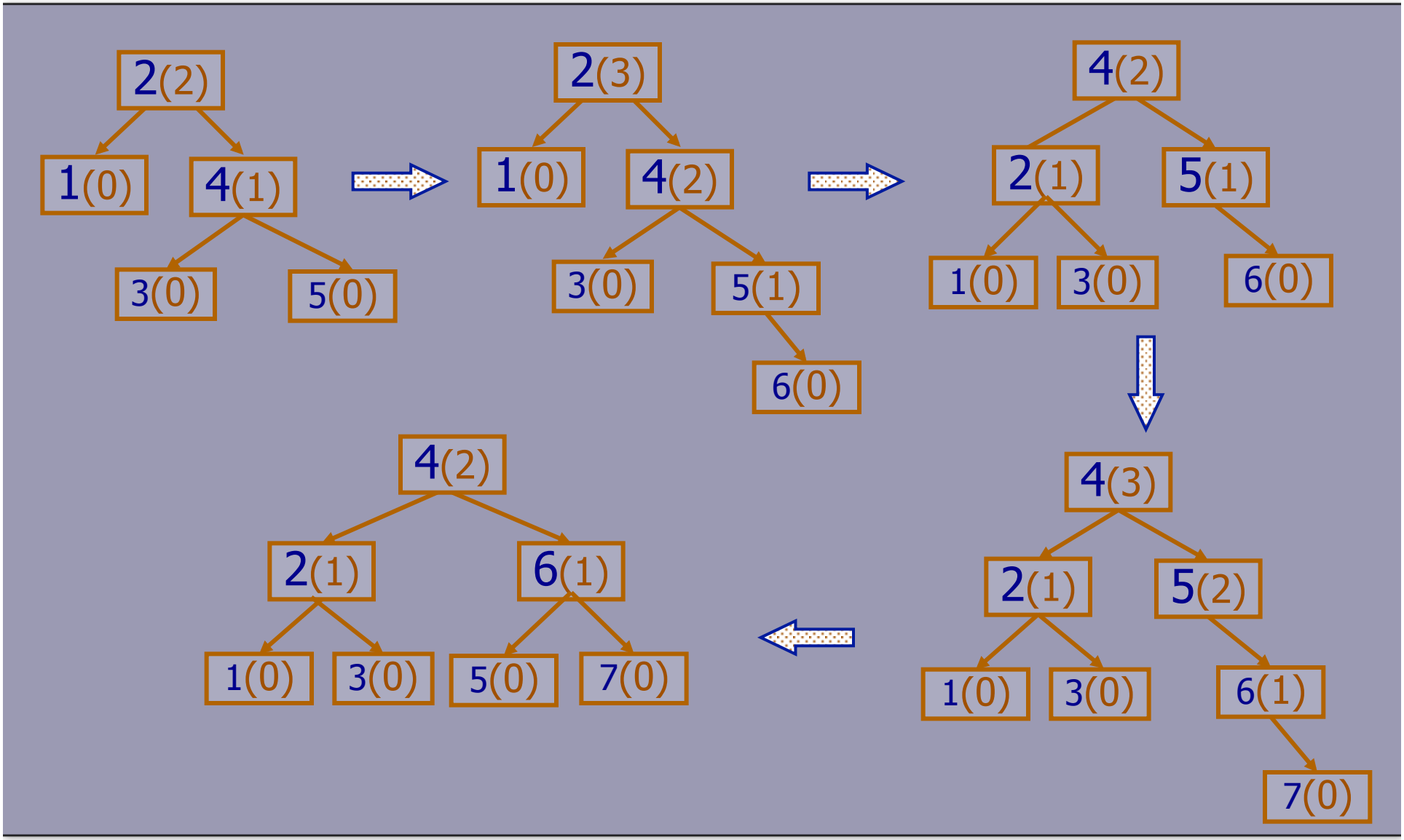


Exercises: Insert 1-7 to an empty AVL tree

- Rebalancing is performed bottom up after a new value has been inserted, and only if the difference in heights of the child trees are more than one.



AVL Trees (cont.)



File Handling in C – File Pointers

C communicates with files using file pointers.
This data type is defined within stdio.h, and written as FILE *

Usage:

```
FILE *output_file;
```

Opening a file pointer

Your program can open a file using the fopen function, which returns the required file pointer.

If the file cannot be opened for any reason then NULL will be returned.

Usage:

```
output_file = fopen("filename.txt",
"w");
if (!output_file)
 fprintf(stderr, "Cannot open %s\n",
"filename.txt");
```

Opening a file pointer (contd.)

fopen takes two arguments, both are strings:

1. the name of the file to be opened ("filename.txt").
2. an access character, which is usually one of:

   "r" : open file for reading

   "w" : open file for writing (create file if it does not exists)

   "a" : open file for appending

Reading/Writing a file

Once the file is opened, you can use the fscanf/ fprintf to read/write to a file.

```
int matched = fscanf(output_file,"%c
%d %s\n", &cmd, &class, name);
fprintf(output_file, "%c \n", cmd);
```

fscanf() returns the number of items in the argument list it can match. Otherwise, EOF is returned.

Reading/Writing a file

EOF is a character which indicates the end of a file.

```
while (fscanf(output_file, "…", …) != EOF)
{ … }
```

EOF is returned by read commands of scanf functions when they try to read beyond the end of a file.

Closing a file pointer

The fclose command is used to disconnect a file pointer from a file.

Usage:

```
fclose(output_file);
```

Systems have a limit on the number of files which can be open simultaneously, so it is a good idea to close a file when you have finished using it.

Standard I/O in C

Standard I/O: input and output channels between a computer program and its environment.

Standard input (stdin): usually input from the keyboard.

Standard output (stdout): usually output to the text terminal (the screen).

Standard error (stderr): to output error messages or diagnostics. Usually output to the screen also.

stdin, *stdout*, *stderr* are 'special' file pointers; don't open or close them.

Standard I/O Functions

Output functions: printf()
Input functions: scanf()

```
int a;
printf("Please enter an integer: ");
scanf("%d", &a);
printf("You typed: %d\n", a);
```

Other input functions: getchar(), fgets(),…

fgets()

An input function (file get string)

```
char* fgets(char *string, int length,
FILE *stream)
```

fgets() reads a string of specific length from a file (or standard input) pointed to by stream.

fgets() terminates reading:

- after a new-line character (\n) is found, OR
- after it reaches end-of-file (EOF), OR
- after (length - 1) characters have been read