Oregon State University
School of Electrical Engineering and Computer Science

CS 261 – Recitation 6

Fall 2012

Outline

- AVL tree
- Assignment 5 – Heap Implementation of a ToDo List
  - Strings in C
  - File Handling and Standard I/O in C

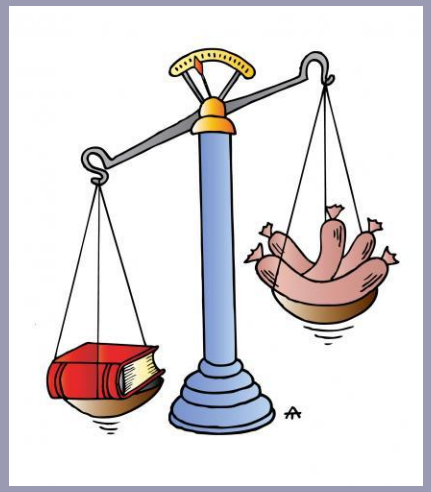
Several parts of this lecture were taken from:
www.cs.txstate.edu/~rp44/cs3358_089/Lectures/bst.ppt

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

AVL Tree is…

- named after **A**delson-**V**elskii and **L**andis

- Binary search tree with **balance condition** in which the sub-trees of each node can differ by at most 1 in their height

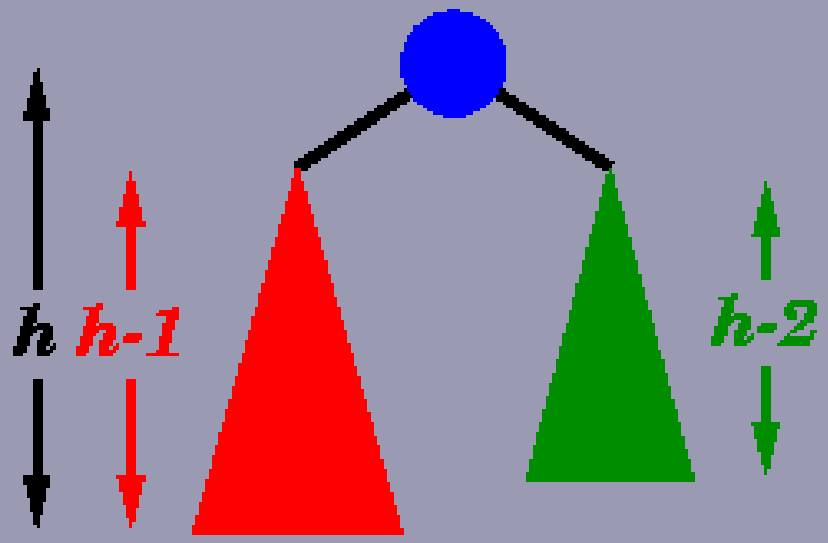


Definition of a balanced tree

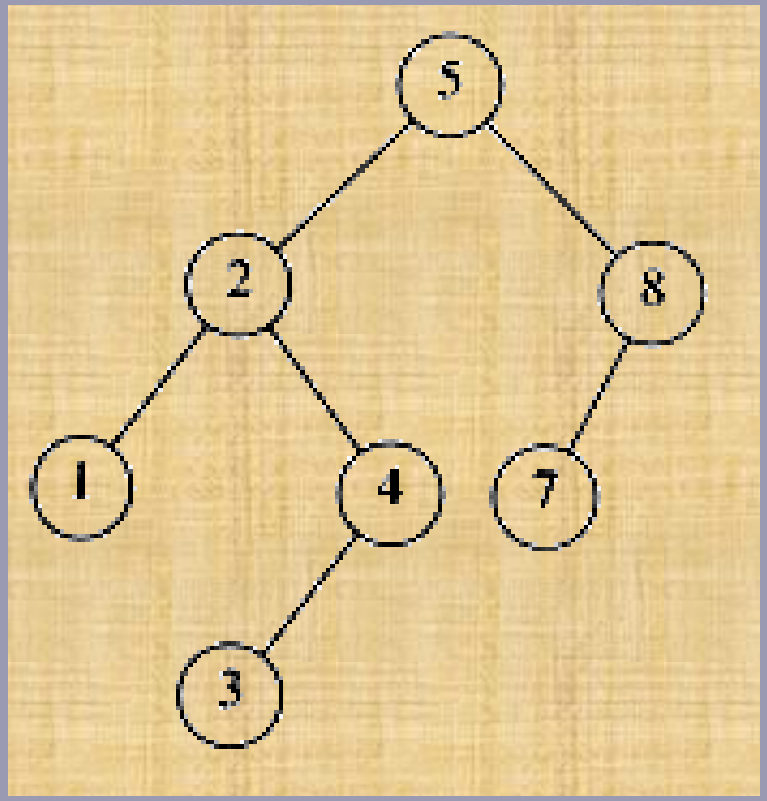
- Ensure the depth = O(log N)
- Take O(log N) time for searching, insertion, and deletion
- Every node must have left & right sub-trees of heights that differ at most 1

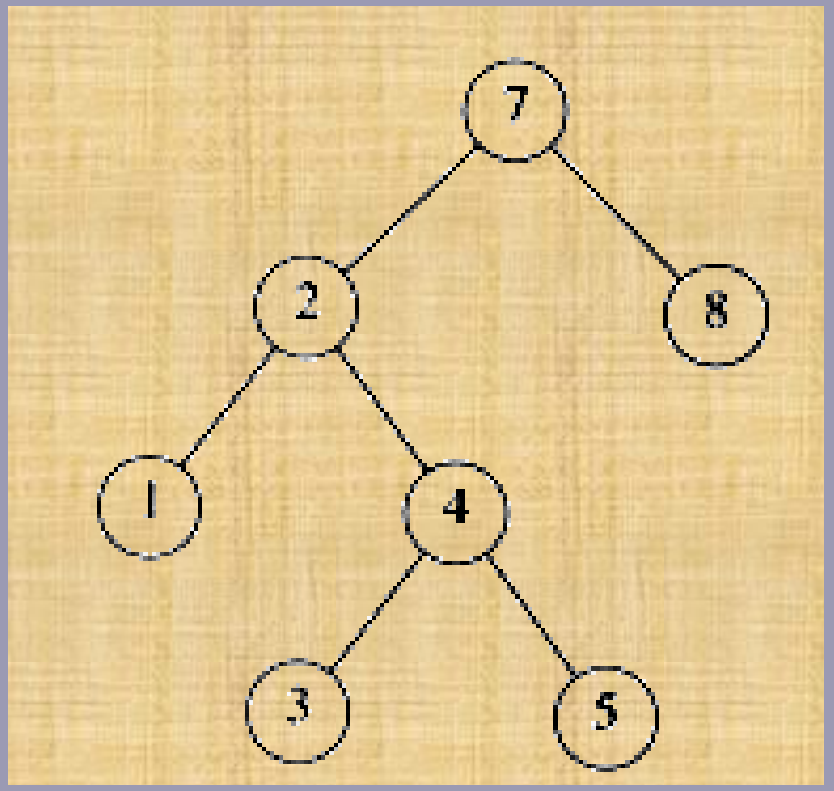
An AVL tree has the following properties

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



AVL tree?





YES

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

NO

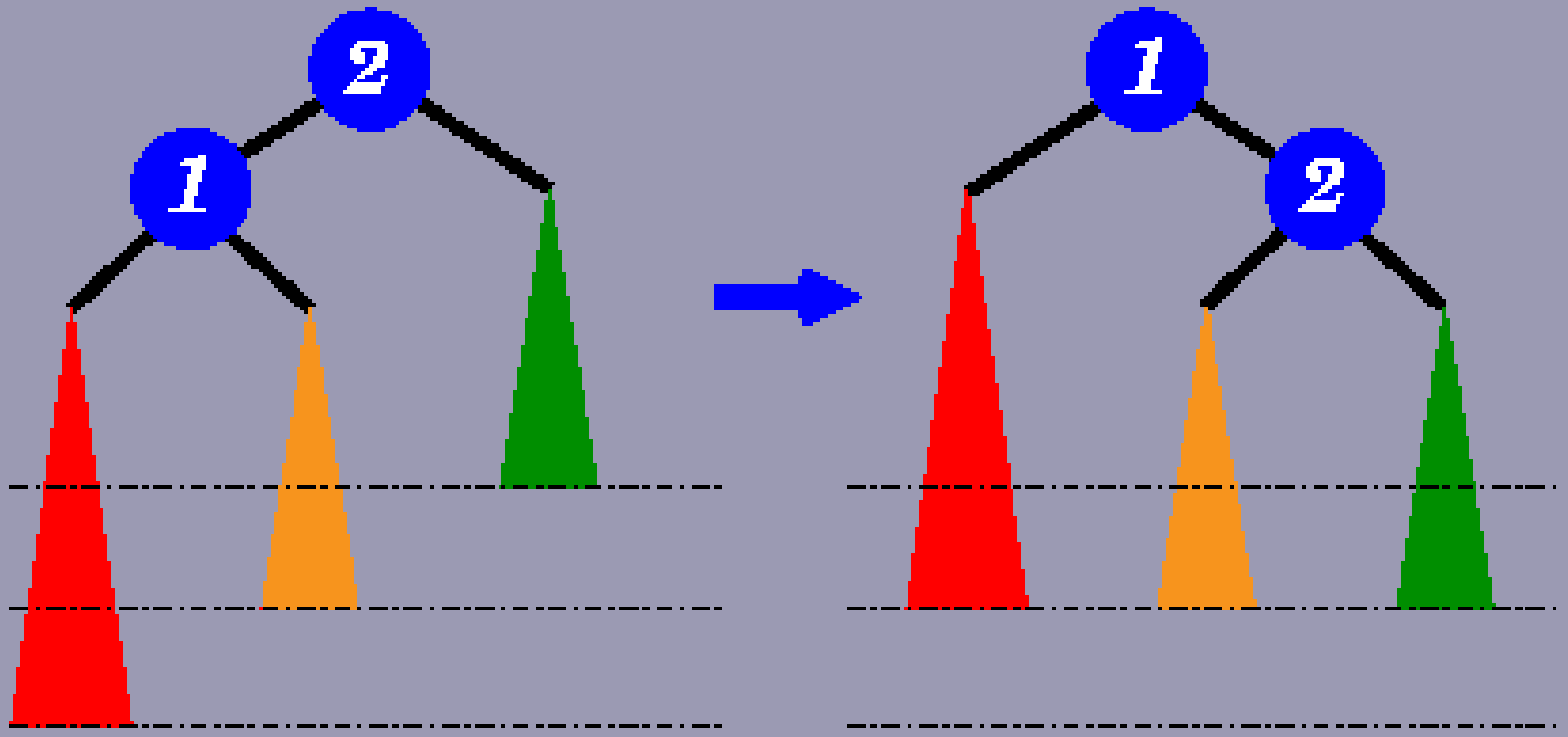
Left sub-tree has height 3, but right sub-tree has height 1

Insertion and Deletions

- It is performed as in binary search trees
- If the balance is destroyed, rotation(s) is performed to correct balance

- For insertions, exactly one single or double rotation is required.
- For deletions, O(log n) rotations at most are needed

Single Rotation



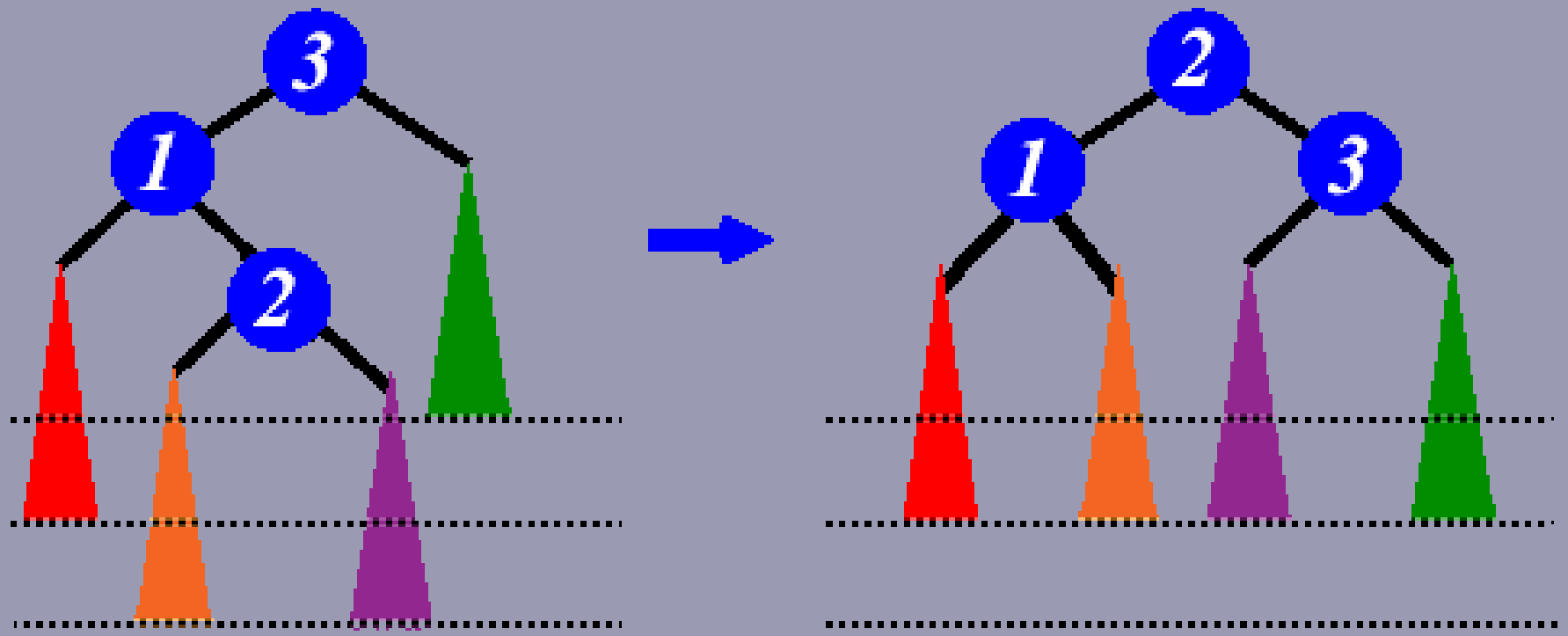
left sub-tree is two level

deeper than the right sub-tree

move ① up a level and

② down a level

Double Rotation



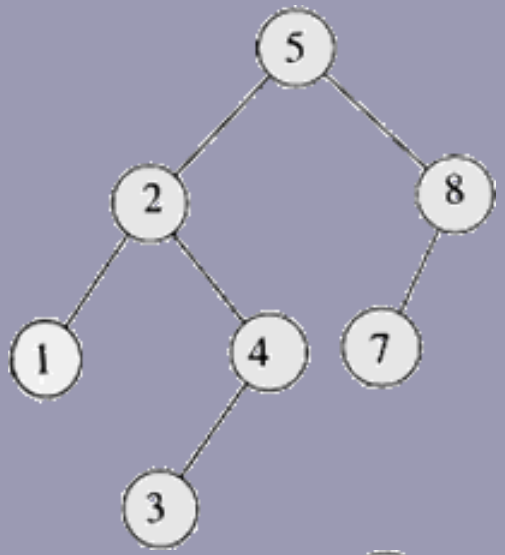
Left sub-tree is two level

deeper than the right sub-tree

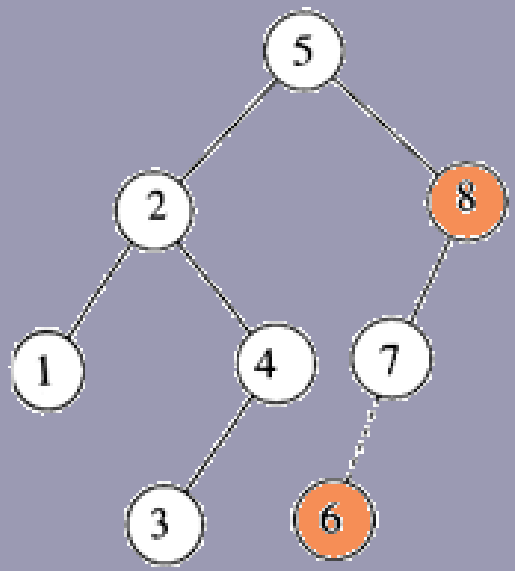
Move ② up two levels and

③ down a level

Insertion

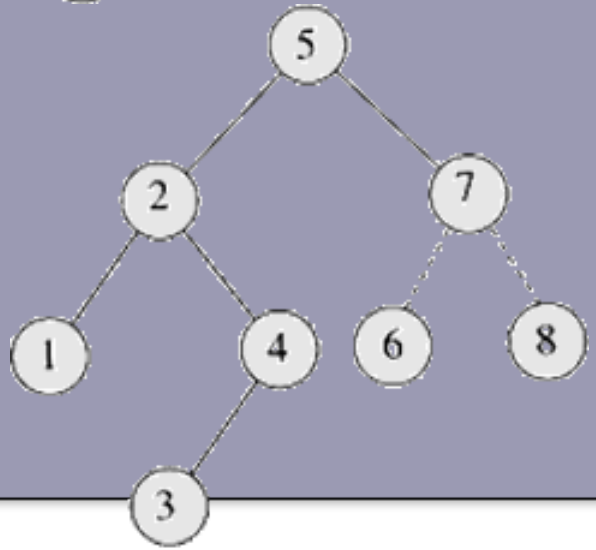


Insert 6

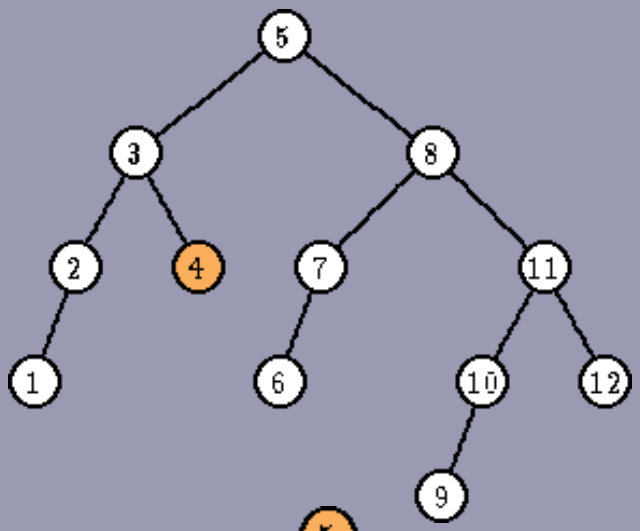


Imbalance at 8

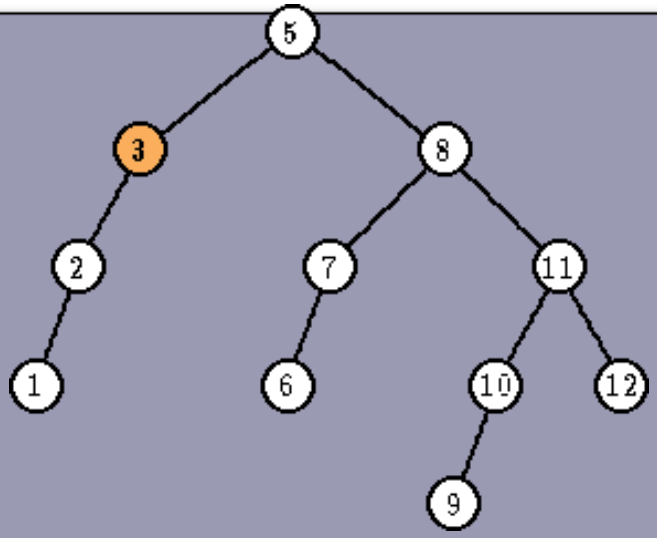
Perform rotation with 7



Deletion

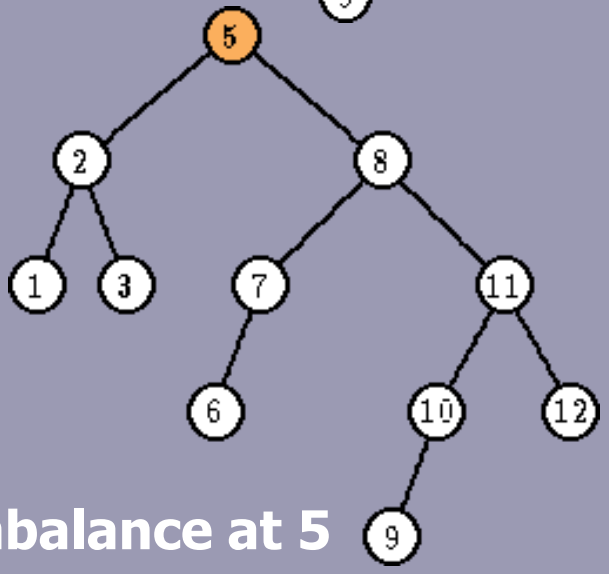


Delete 4

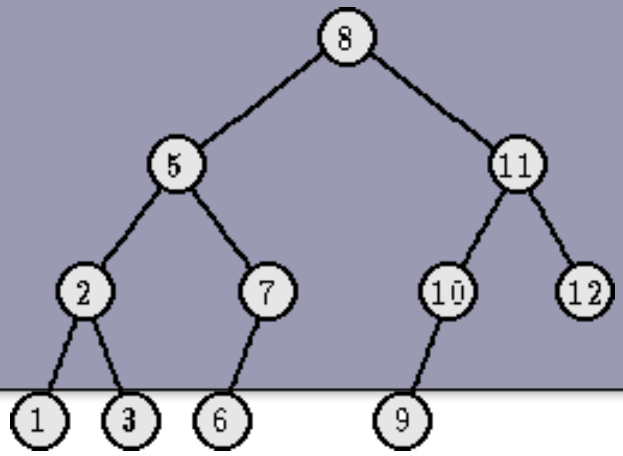


Imbalance at 3

Perform rotation at 3



Imbalance at 5



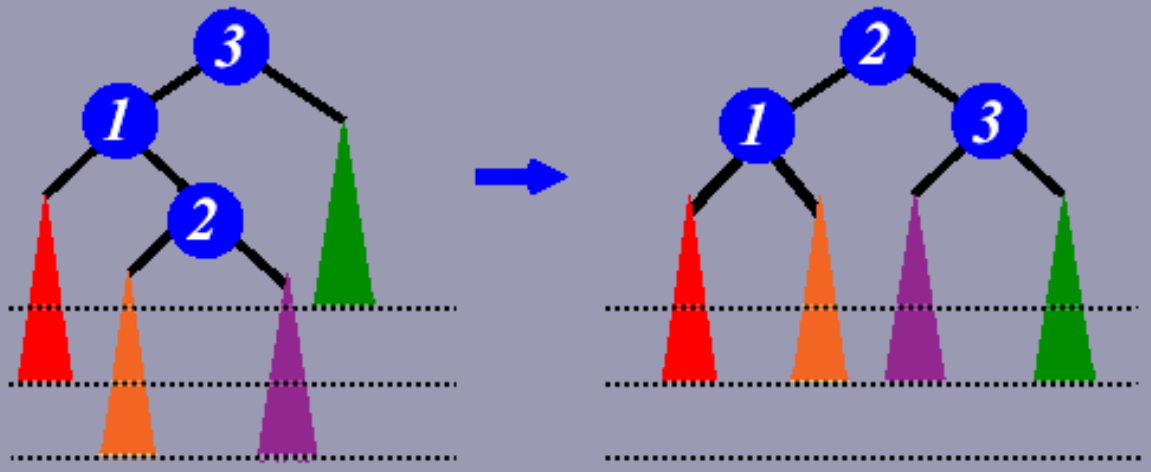
Key Points

- 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?

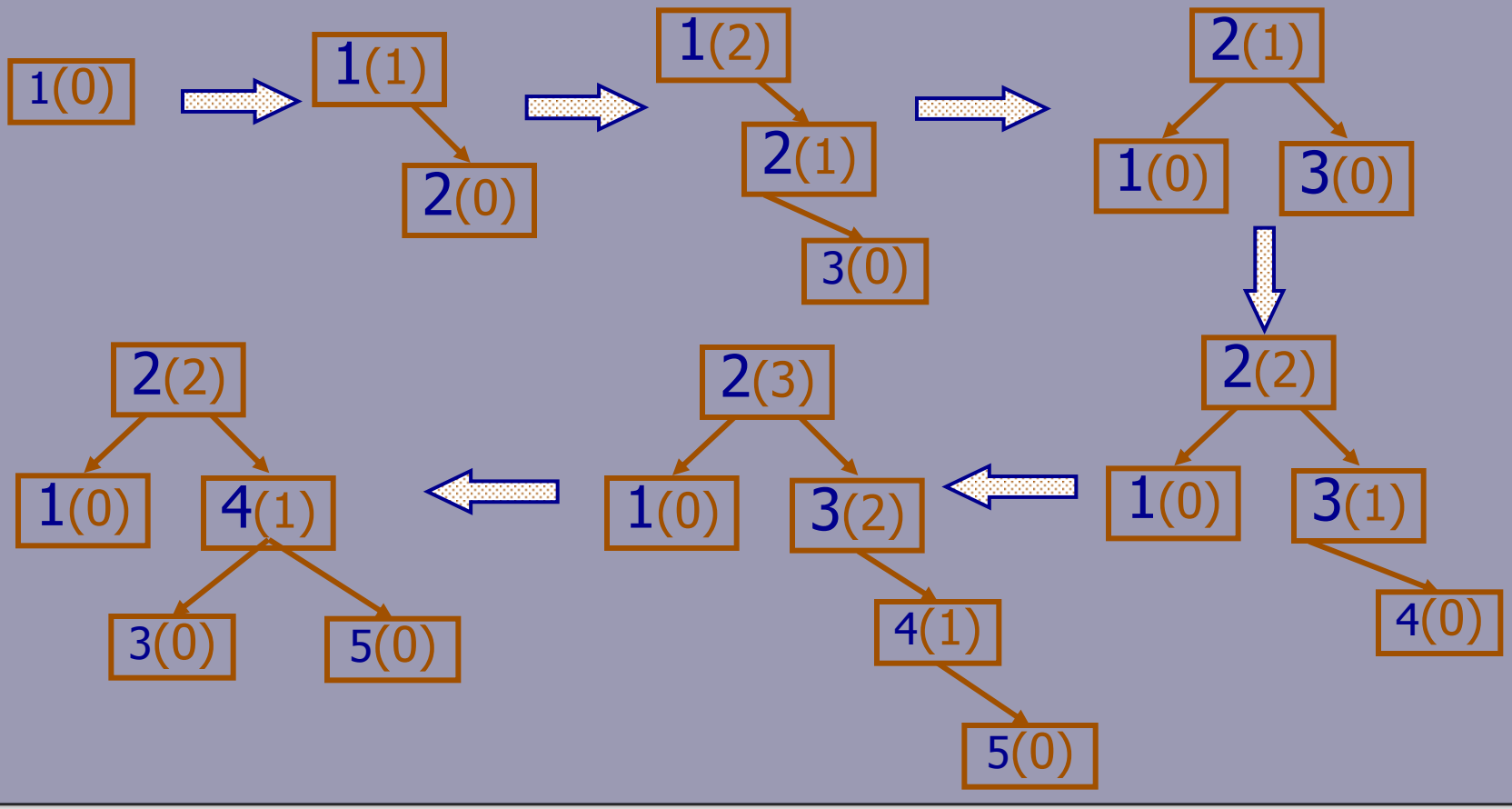
Balance Factor = Height(left subtree) – height(right subtree)

- At an unbalanced node N, a double rotation is needed when:
  - N's BF is positive and N's left subtree's BF is negative
  - N's BF is negative and N's right subtree's BF is positive.

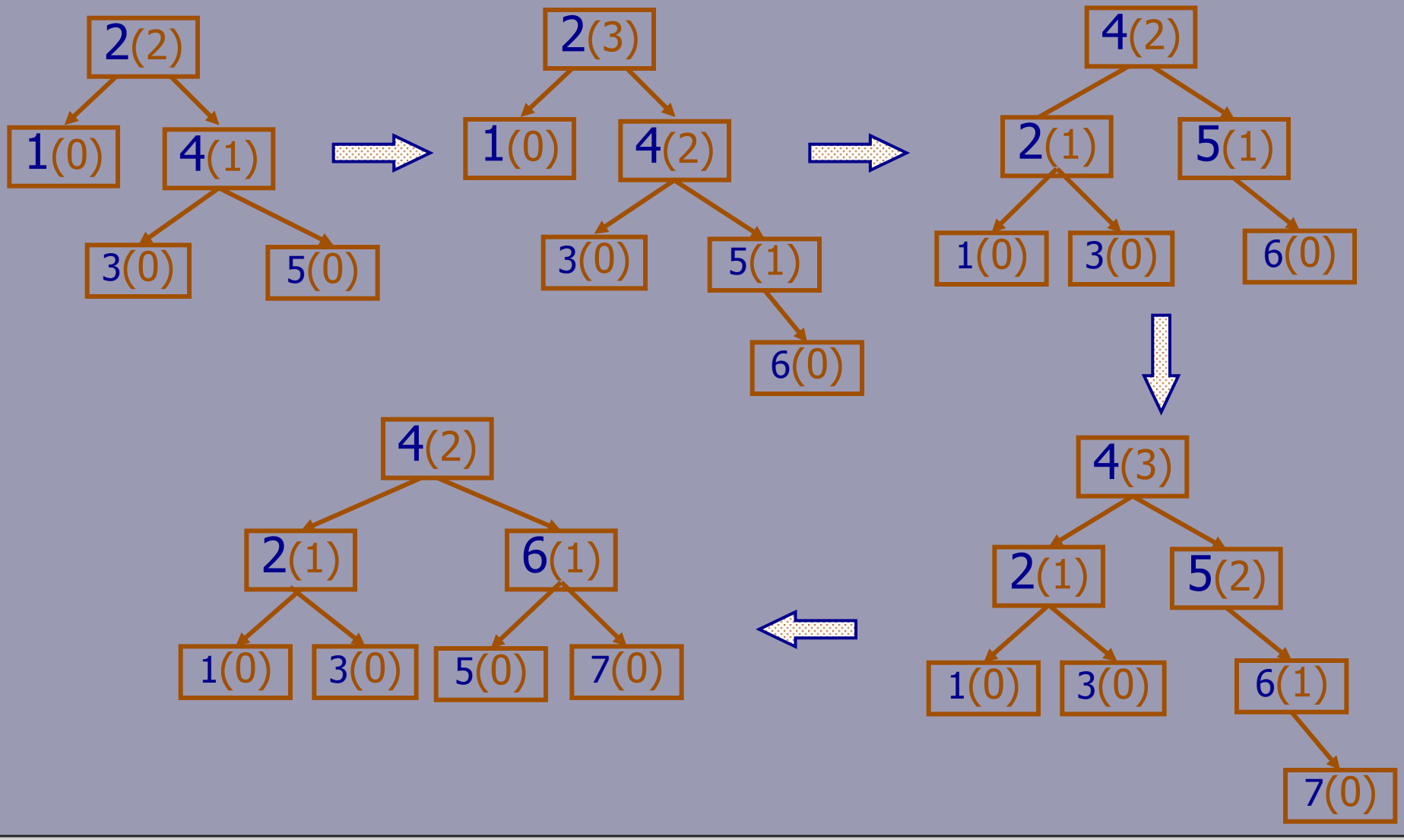


Exercises: Insert 1-7 to an empty AVL tree

- Remember that rebalancing AVL trees 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.)



Strings in C

In C, a string is an array of characters terminated by a null character (\0).

File Handling in C – File Pointers

C communicates with files using a new datatype called a file pointer.

This 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 the value NULL will be returned.

Usage:

```
output_file = fopen("filename.txt", "w");
if (output_file == NULL)
 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.

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

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

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

EOF is returned by read commands of scanf families 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.

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