Lab 3

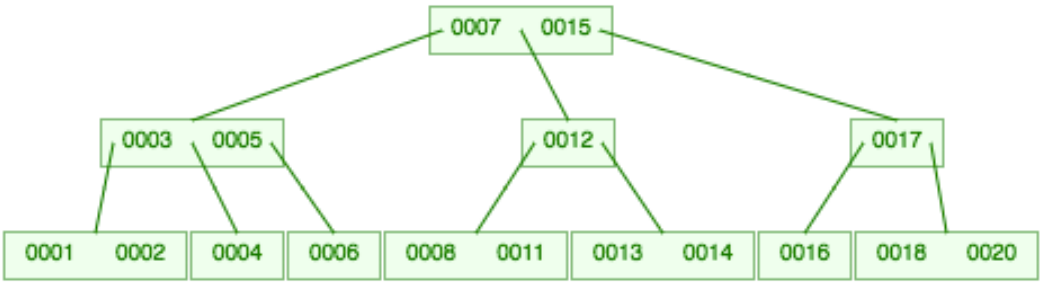
Advanced Databases

Name: Eryk Szlachetka
Student Number: C14386641
Course Code: DT282/4
Lab: 3

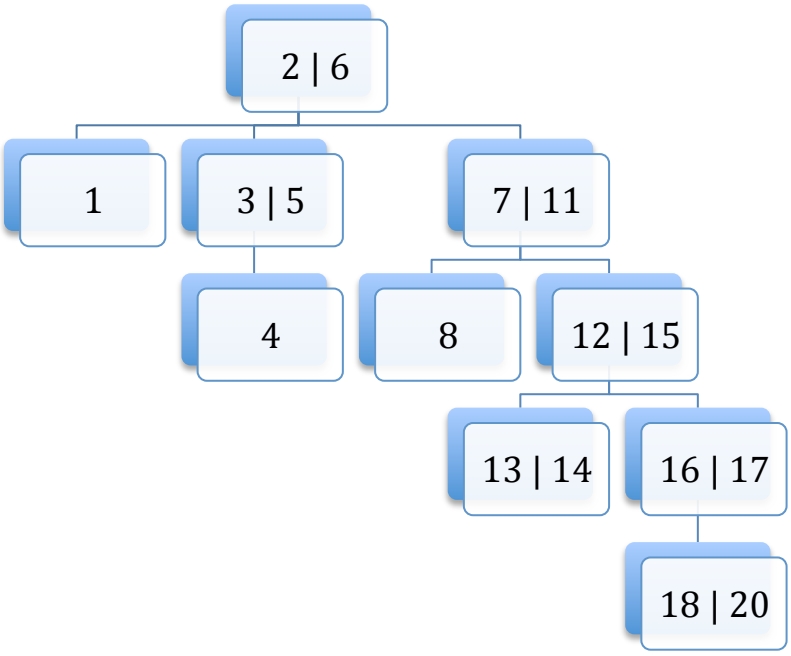
C14386641@mydit.ie

Exercise 1

2, 6, 5, 7, 11, 3, 12, 15, 13, 16, 4, **1**, 8, 14, **17**, 18, 20



Exercise 2



What is the main difference? What is the main consequence?

The main difference is that the tree is unbalanced meaning that it continues to one side only (right). The consequence is that the performance goes down.

Exercise 3 – Bitmap Index

- What is a bitmap index?

A **bitmap index** is a special kind of database **index** that uses bitmaps. **Bitmap indexes** have traditionally been considered to work well for low-cardinality columns, which have a modest number of distinct values, either absolutely, or relative to the number of records that contain the data.

- How does it work?

Bitmap indexes use bit arrays (commonly called bitmaps) and answer queries by performing bitwise logical operations on these bitmaps.

- What are the advantages and disadvantages?

Bitmap indexes have a significant space and performance advantage over other structures for query of such data. Their drawback is they are less efficient than the traditional B-tree indexes for columns whose data is frequently updated: consequently, they are more often employed in read-only systems that are specialized for fast query - e.g., data warehouses, and generally unsuitable for online transaction processing applications.

- How big (in Mbytes) is a Bitmap index for a field "Month" (12 values) of a table of 5 million records?

The Bitmap is going to have 1 bit for each record, representing the month i.e.

ID	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1	**1**	0	0	0	0	0	0	0	0	0	0	0
2	0	**1**	0	0	0	0	0	0	0	0	0	0
..	..	..	..	..	..	..	..	..	..	..	..	..
5 mln	0	0	0	0	0	**1**	0	0	0	0	0	0

Therefore **5,000,000 bits * 12 rows** = 60,000,000 bits.
60,000,000 bits = **7.15 MB**

Exercise 4

- The average number of nodes that needs to be visited for a b-tree to check if a number is:

  2.5 (23% out of 11 nodes)

- The average number for the simple tree is

  2.9 (29% out of 10 nodes)

- The gain in performance:

  The b-tree is 6% faster because it needs to visit on average 6% less number of nodes.