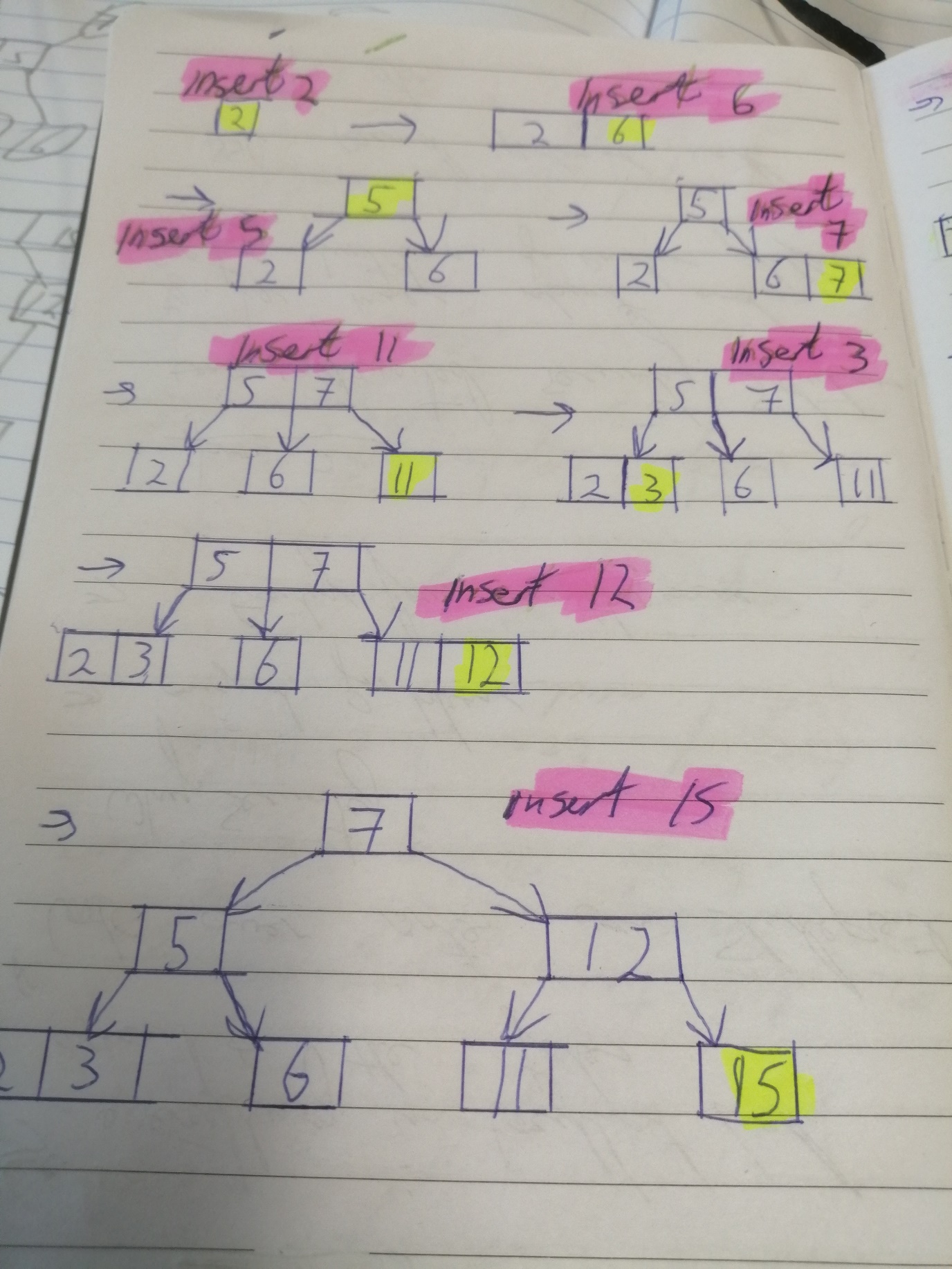
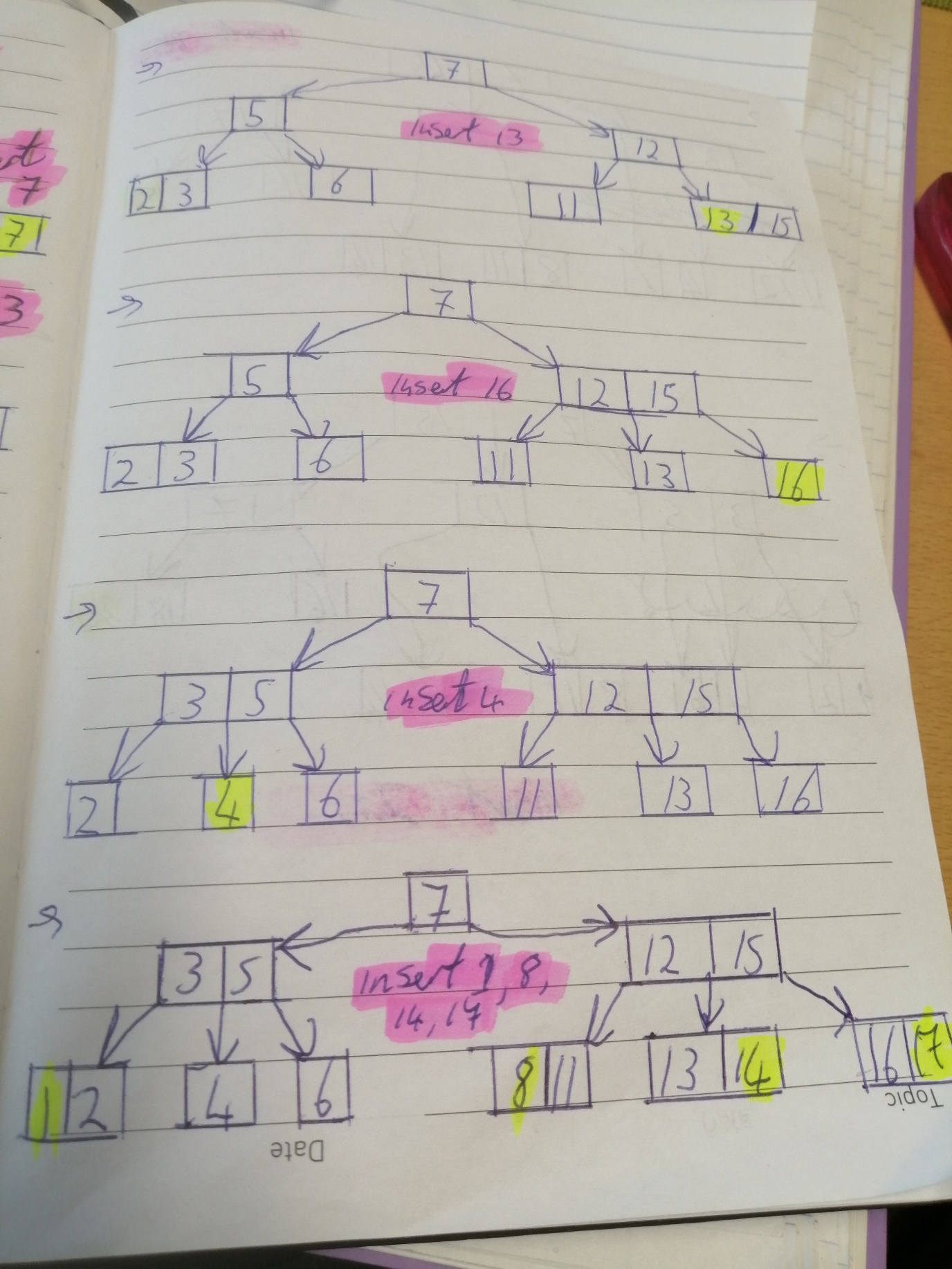
Lab 3 Trees

# B Tree

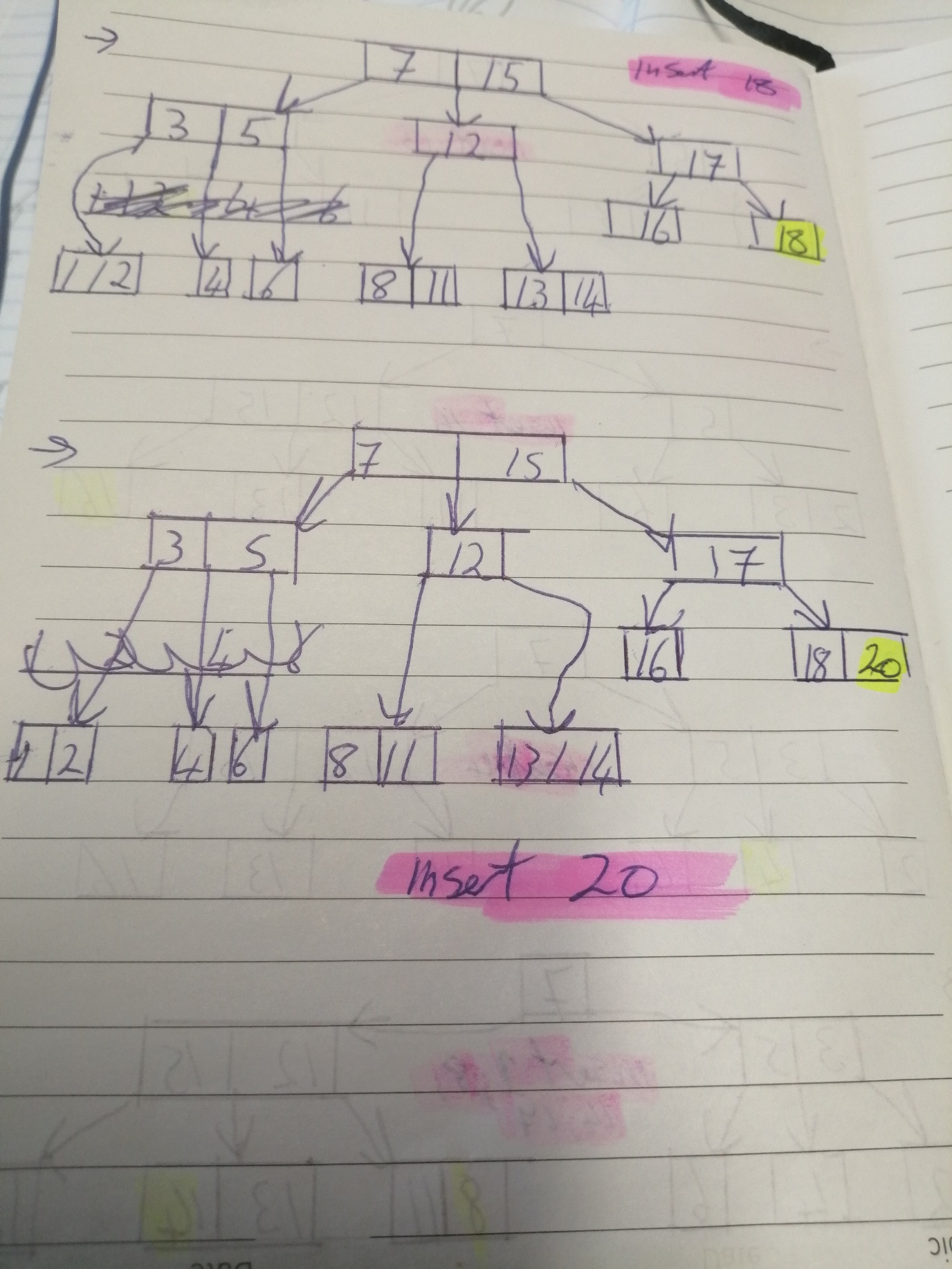
Part 1



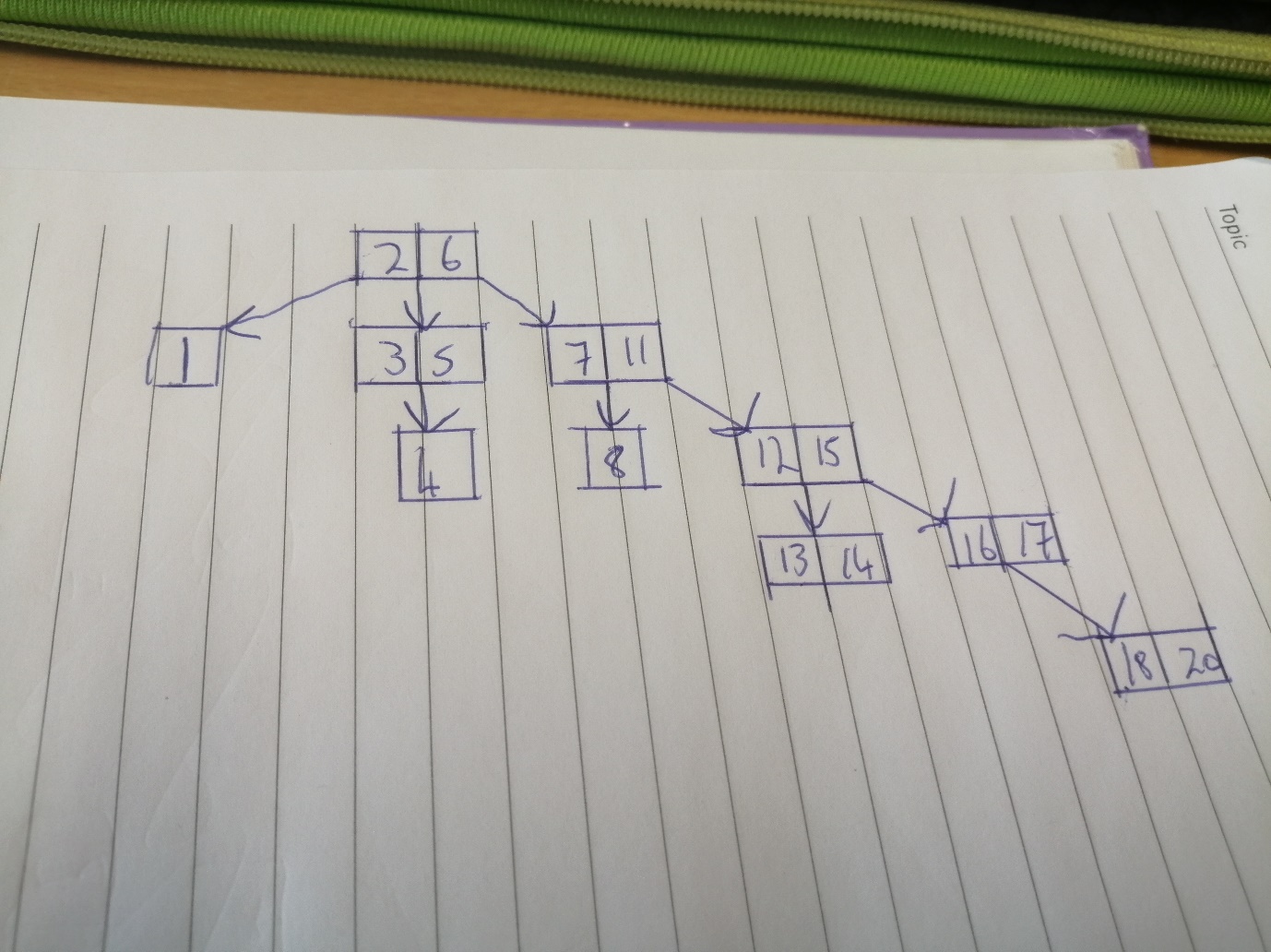
Part 2



Part 3



# STree



Answer to the question

The main difference between the two is as the result of the none balancing rule, the tree skews to the right side.

The main consequence is the searching pattern, for the bigger numbers, would take longer time with more memory management as you go deeper into the levels. The breath-first search algorithm is also not ideal with this model design as oppose to the rule of the balancing tree model.

# Problem 3

A bitmap index is a database index that uses bitmaps or bit array.

A bitmap index functionality is to store a bitmap for each index key, which is used to store to store pointers to multiple rows.

The advantage of such system is we can use the functionality to read a collection of data from a schema and understand the meaning behind the data combined with the amount of memory used to store it combined with time processed to collect the requested data.

The disadvantage is time management when the cardinality of the data is medium to high. The accuracy can also be a problem with tables that are frequently updated. Best practises is a combination of using bitmap indexes for read-only table with low cardinality.

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

1 Byte = 8 bits. 12 values = 12 bits = 1.5 bytes. 1.5 \* 5000000 (5 million) = 7500000 (7.5 million) = 7.5Mbytes

# Problem 4

The average amount of steps to find a number in the b tree is 2.6

The average amount of steps to find a number in the s tree is 2.8

Therefore, the b tree is 7.69% (round to 2 decimal place) faster than the s tree

