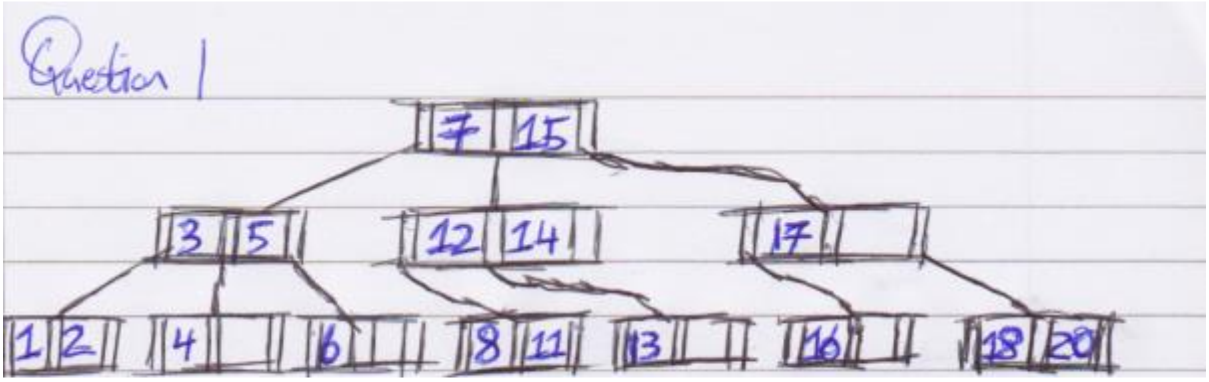
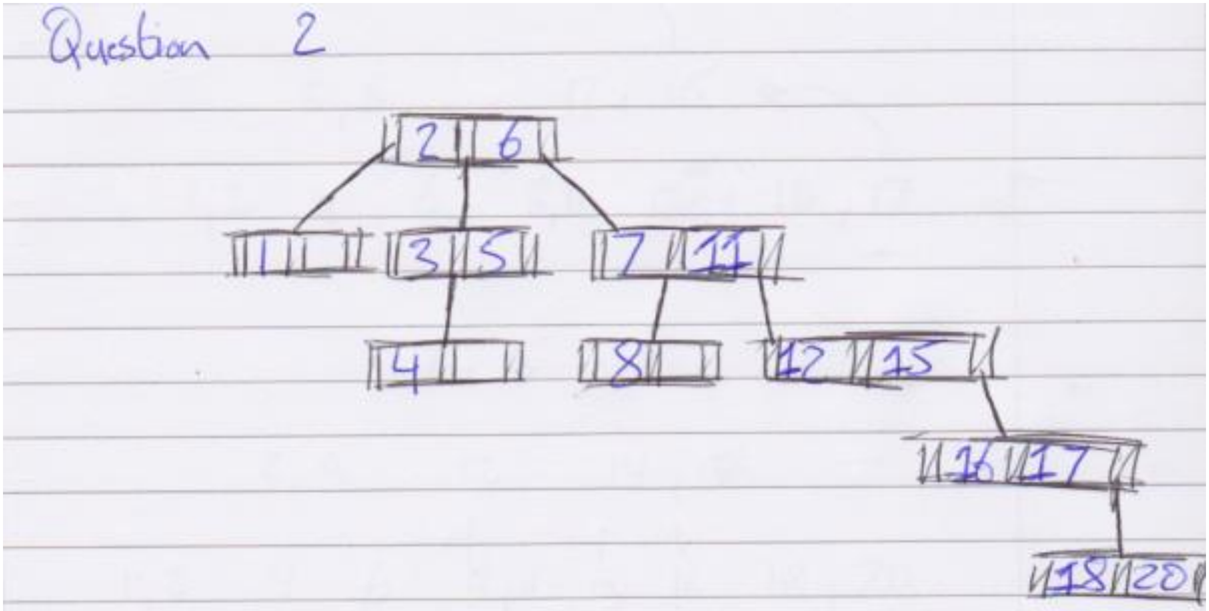
Exercise 1:



Exercise 2:

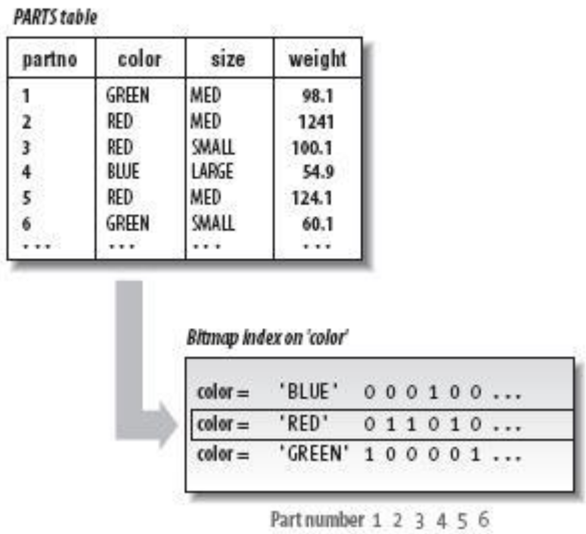


(**Note, there is a node missing in between 12 and 15. The missing node should contain 13 and 14**).

The main difference between the two is how balanced the trees are. In the first exercise, the tree is very balanced. This is because the child nodes are well spread out. In the second exercise however, the tree is very unbalanced, and each node can only have 2 children.

Exercise 3:

A bitmap index is a two-dimensional array where one column for every row in the table is indexed. Each column will represent a distinct value within the bitmapped index. For example:



Advantages:

- Highly compressed structure
- Fast disk reads

Disadvantages:

- High overhead associated with maintenance
- Potential for deadlock

12 options x 5 million records = 60,000,000 bits / 7,500,000 bytes / 7.5 megabytes

Exercise 4:

A) The average number of nodes that needs to be visited for a b-tree to check if a number is in the index or not.

   Answer: Average depth = sum of all depths / number of nodes = 25 / 17 = **1.47**

B) The average number for the simple tree

   Answer: Average depth = sum of all depths / number of nodes = 33 / 17 = **1.94**

C) The gain in performance (for instance “the b-tree is 10% faster because it needs to visit on average 10% less number of nodes”).

   Answer: Average for question 1 / average for question 2 * 100 = percentage difference of average depth = 1.47 / 1.94 * 100 = Question 2 has on average **75.77 %** more depth than question 1.