Q1:

We can assume that node A is node B’s successor. As it is a binary search tree, so node A must be node B’s right child. And node A is the least value greater than node B, so in node B’s right child, there is no node smaller than node A and bigger than node B, which means node A don’t have a left child.

Q2:

2.1: A complete binary tree is a binary tree in which every level (except the last possible level) is filled, and all nodes of the last level are as far to the left as possible. That is, for the last layer, there may not be nodes at all locations, but the nodes that do exist are on the left.

2.2: A heap is a special kind of complete binary tree. In the maximum heap, the value of each parent node is larger than or equal to the value of its children. In the minimum heap, the value of each parent node is smaller than or equal to the value of its children.

2.3：As it is a max heap, so the max value is the top of the heap, and we start at the top. According to the rule, the pointer is maintaining on the largest value, so it will be always swap with its larger child, so it is always going down the tree until it doesn’t have a child.

2.4: no, the new tree may not be a complete tree, in this way, at the last row of the tree, it may have just one node in left child and one node in right child.

2.5: delete the top node(max value in the max heap), then move the last value in this tree to the top, maintain a pointer to the new top value and swap it with the child of larger value until it has no child.

Q3:

3.1:

if capacity is 12:

* First day, [1,2,3,4]
* Second day, [5,.6]
* Third day, [7]
* Fourth day, [8]
* Fifth day, [9]

Can not load all packages.

If capacity is 17:

* First day, [1,2,3,4,5]
* Second day, [6,7]
* Third day, [7,8]
* Fourth day, [9]
* Fifth day, [10]

Loaded all packages.

3.2:

if the capacity of the truck is equal to the sum of the weights of the packages, we can load all packages in one day.

3.3: