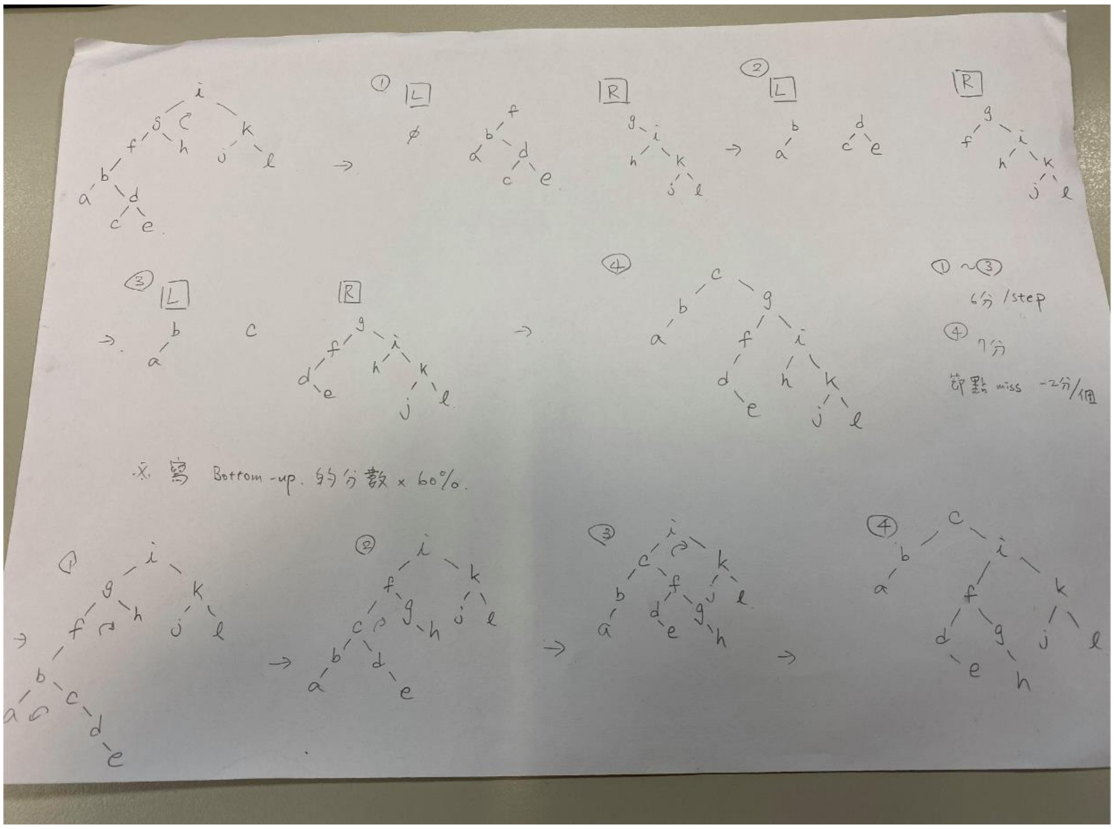
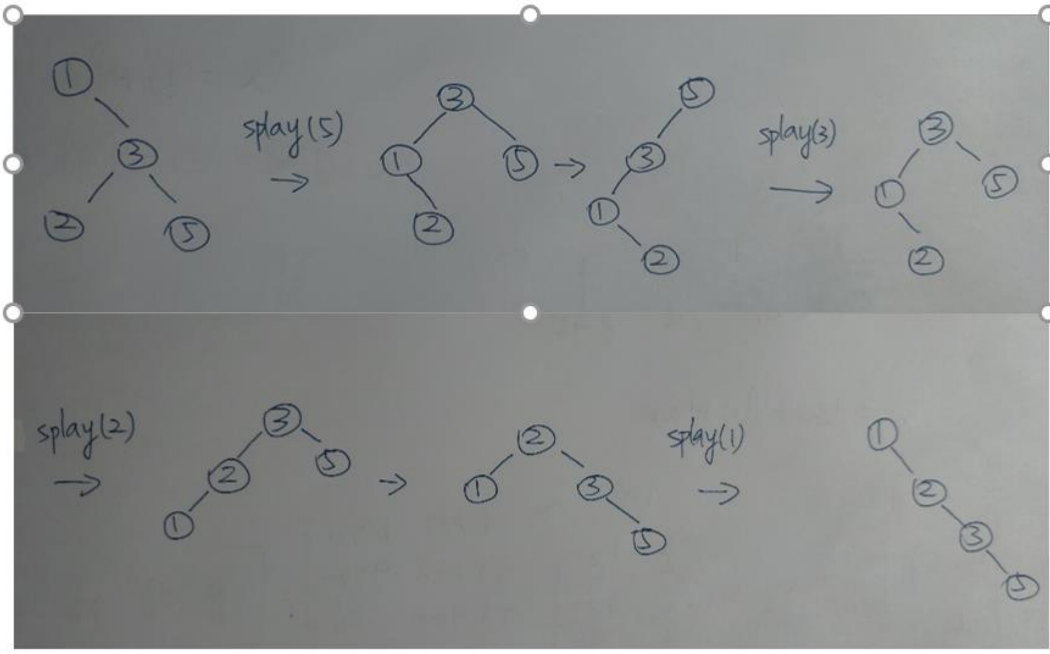
資料結構 HW3 解答

Q1:



Q2:

This question can be solved by Splaying from n, n-1 ... to (1).



Q3:

Let potential function $\emptyset_i$ denote the number of elements in the stack at operation i.

For push operations, amortized cost is counted as below:

$$a_i = 1 + \emptyset_{i+1} - \emptyset_i = 2$$

For multi-pop operations:

$$\begin{aligned} a_i &= k + \emptyset_{i+1} - \emptyset_i \\ &= k + \emptyset_i - k - \emptyset_i \\ &= 0 \end{aligned}$$

Total:

$$\sum_{i=1}^{n} a_i{}' = \sum_{i=1}^{n} a_i + (\emptyset_n - \emptyset_0)$$

Because the term $(\emptyset_n - \emptyset_0)$ is greater than zero:

$$\sum_{i=1}^{n} a_i \leq 2n$$

N operations will take at most 2n time.

Q4:

Finding the heights of T and U (h_T and h_U) needs O(log n) and O(log m), respectively.

If $h_T < h_U$:

Step1: Traversing T from root to find the right-most item(maximum).

Step2: Set it as x and delete it from T.

Now, T becomes T_1.

Step3: Insert x into the left node of U at the height of h_T+1.

If the node has already 4 elements, split it by the insertion rule.

Step4: Set T_1 as a left-subtree of that node.

Done.

If h_T >= h_U, do the opposite operation.
Follow the steps above by changing "right" and "left" to each other.

Step1: Traversing U from root to find the left-most item(minimum).

Step2: Set it as x and delete it from U.

U becomes U_1.

Step3: Insert x into the right node of T at the height of h_U+1.

Step4: Set U as a right- subtree of that node. Done.