1.

We want to multiply 2101 and 1130. We have

$$\begin{aligned} c_2 &= 21 * 11 \\ c_0 &= 01 * 30 \\ c_1 &= (21 + 01) * (11 + 30) - ((21 * 11) + 01 * 30) \\ &= (22 * 41) - (21 * 11) - (01 * 30) \end{aligned}$$

Thus we break it down further as follows.
For $21 * 11$ we have:

$$\begin{aligned} c_2 &= 2 * 1 = 2 \\ c_0 &= 1 * 1 = 1 \\ c_1 &= (2 + 1) * (1 * 1) - (2 * 1 + 1 * 1) \\ &= 3 \end{aligned}$$

So

$$21 * 11 = 2 \cdot 10^2 + 3 \cdot 10 + 1 = 231$$

For $01 * 30$ we have:

$$\begin{aligned} c_2 &= 0 * 3 = 0 \\ c_0 &= 1 * 0 = 0 \\ c_1 &= (0 + 1) * (3 + 0) - (0 + 0) \\ &= 3 \end{aligned}$$

So

$$01 * 30 = 0 \cdot 10^2 + 3 \cdot 10 + 0 = 30$$

For $01 * 30$ we have:

$$\begin{aligned} c_2 &= 2 * 4 = 8 \\ c_0 &= 2 * 1 = 2 \\ c_1 &= (2 + 2) * (4 * 1) - (8 + 2) \\ &= 10 \end{aligned}$$

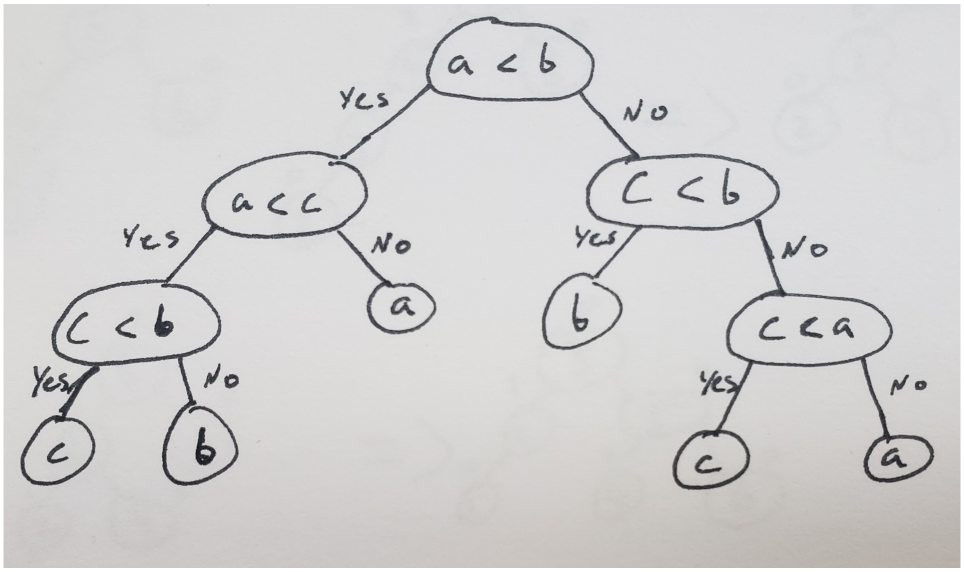
So

$$22 * 41 = 8 \cdot 10^2 + 10 \cdot 10 + 2 = 902$$

Putting it all together, we have

$$\begin{aligned} 2101 * 1130 &= 231 \cdot 10^4 + (902 - 231 - 30) \cdot 10^2 + 30 \\ &= 2,374,130 \end{aligned}$$

2.



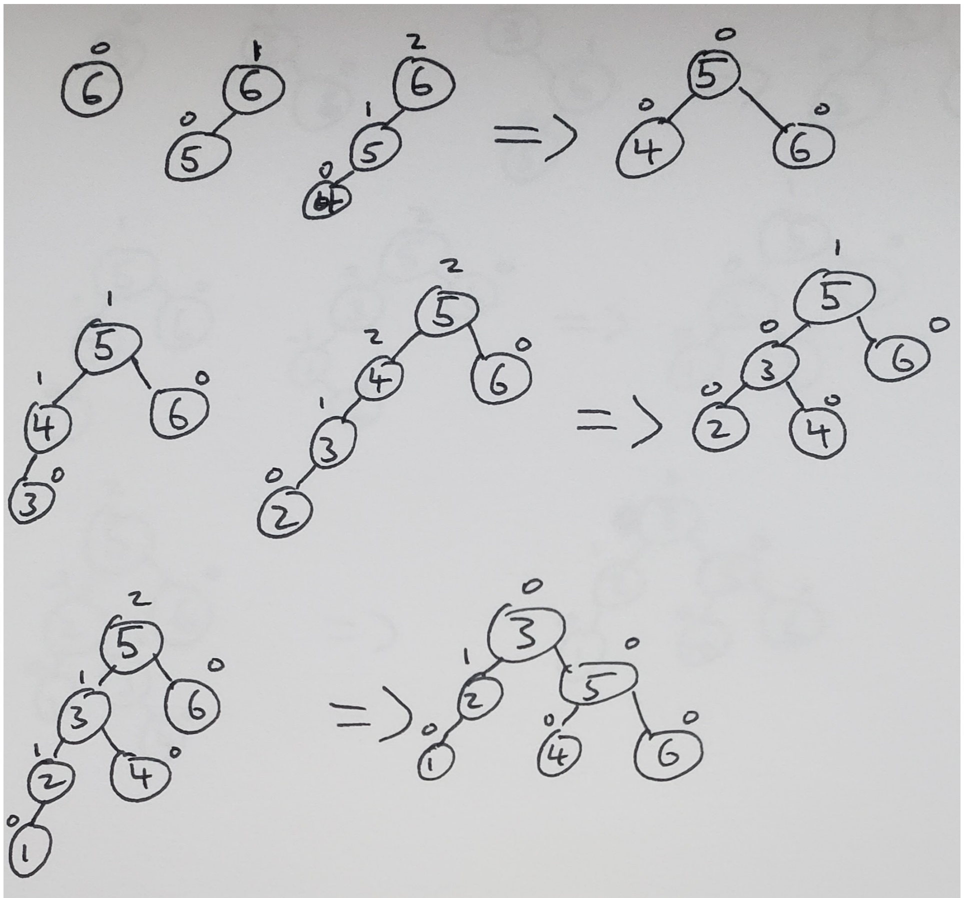
3.

(a) First we sort the array with a sorting algorithm with optimal efficiency will be in the first and last positions of the array respectively. Thus, we have an efficiency as follows:

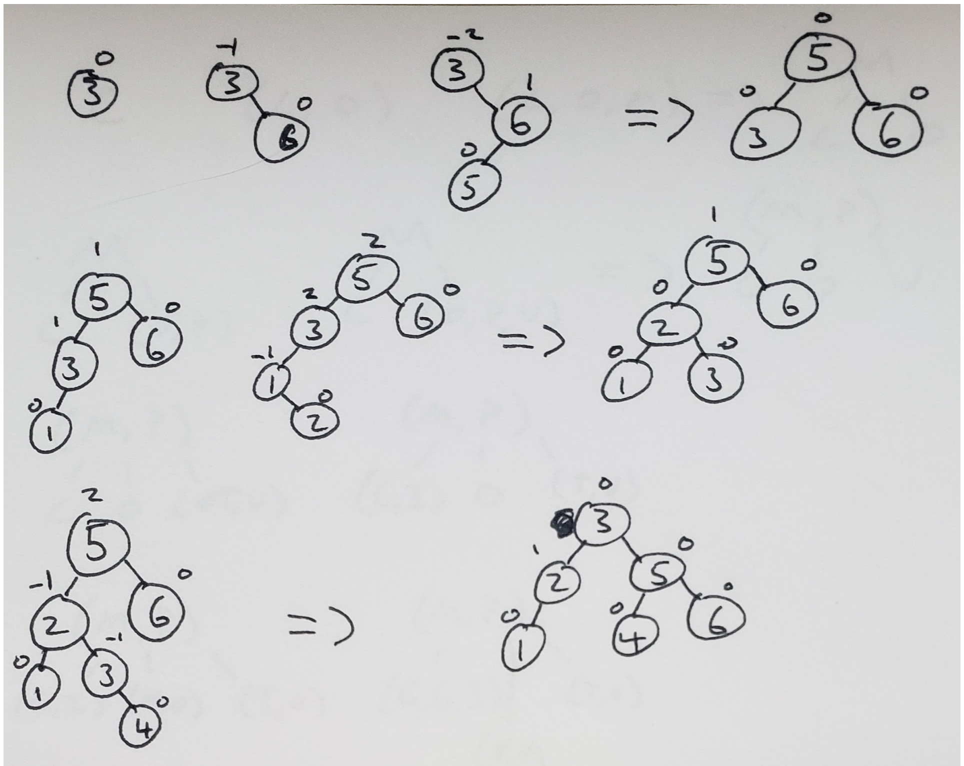
$$T(n) = T_{\text{sort}}(n) + T_{\text{select}}(n) \in \Theta(n \log n) + \Theta(1) = \Theta(n \log n)$$

(b) First assign a varible called `min` to $A[0]$ and then loop through the array and if $A[i]$ is less than `min`, then reassign `min` to $A[i]$. Then create a variable called `max` and loop through again reassigning if an element of the array is larger than `max`. Since we had to loop through the array twice, then $T(n) = 2n$ and $T(n) \in \Theta(n)$.

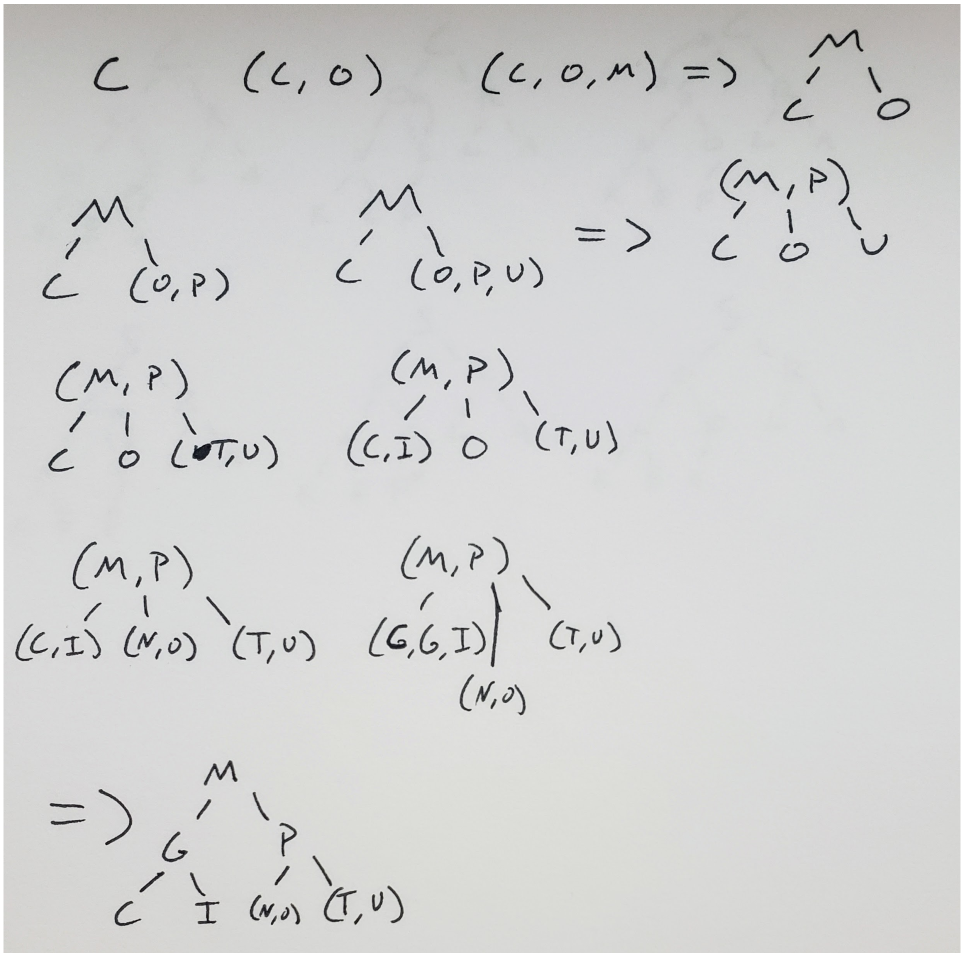
4 (b).



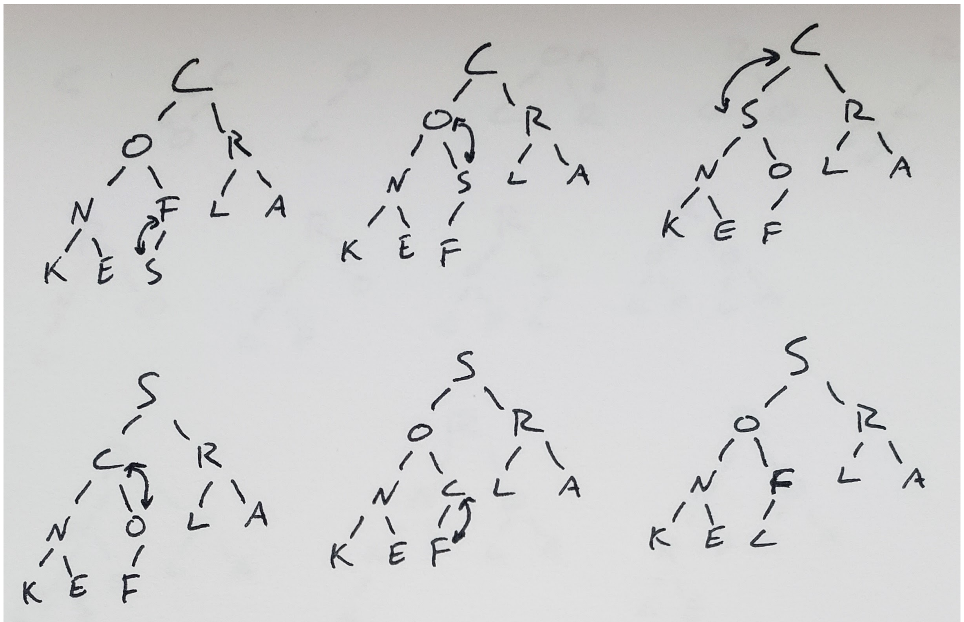
4 (c).



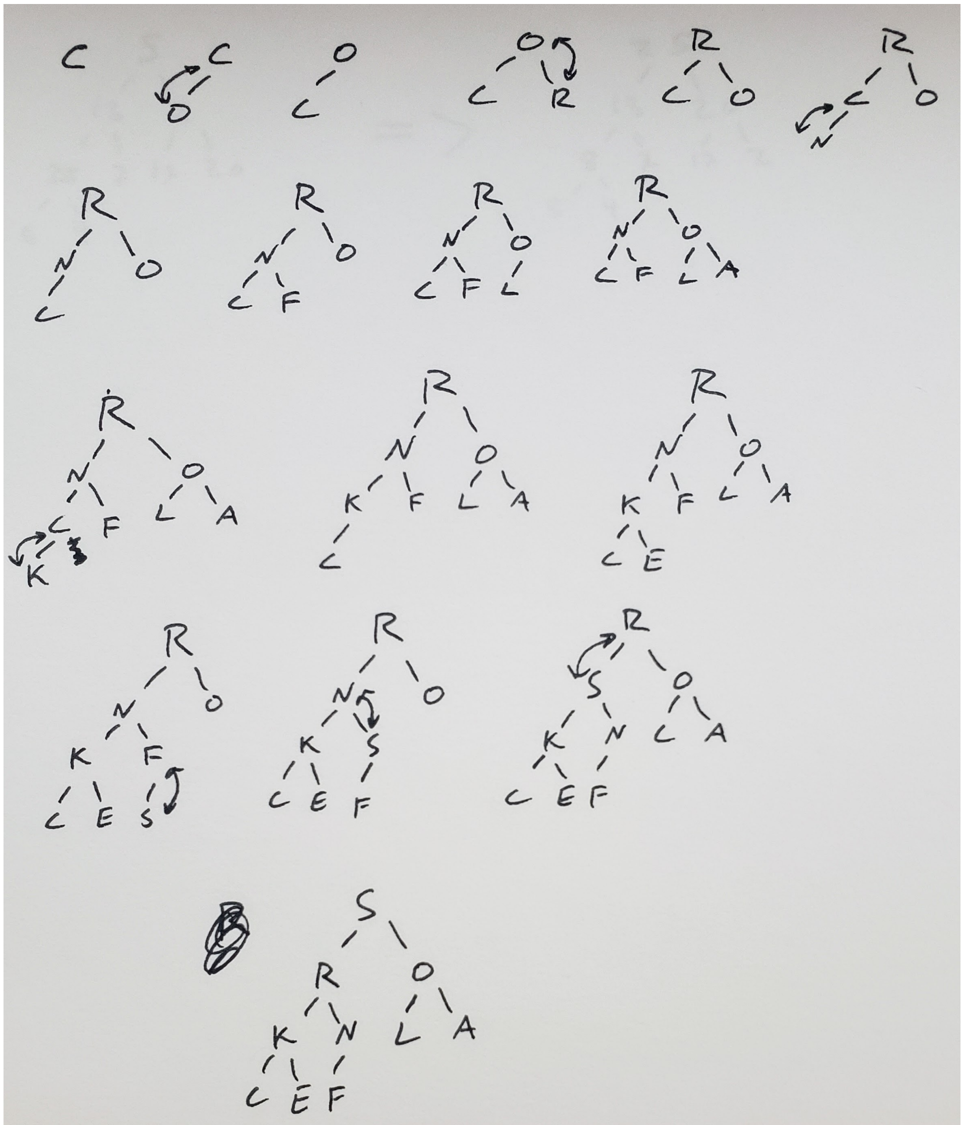
5.


C (C, O) (C, O, M) =>
M
C O
M
C (O, P)
M
C (O, P, U) =>
(M, P)
C O U
(M, P)
C O (T, U)
(M, P)
(C, I) O (T, U)
(M, P)
(C, I) (N, O) (T, U)
(M, P)
(G, G, I) (T, U)
(N, O)
=>
M
G P
C I (N, O) (T, U)

6(a).

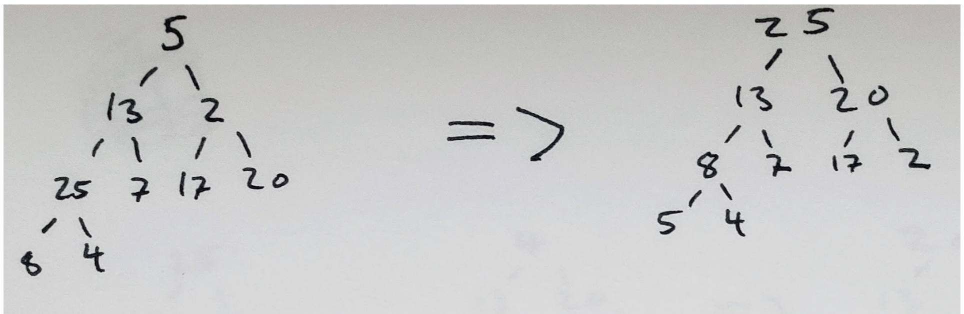


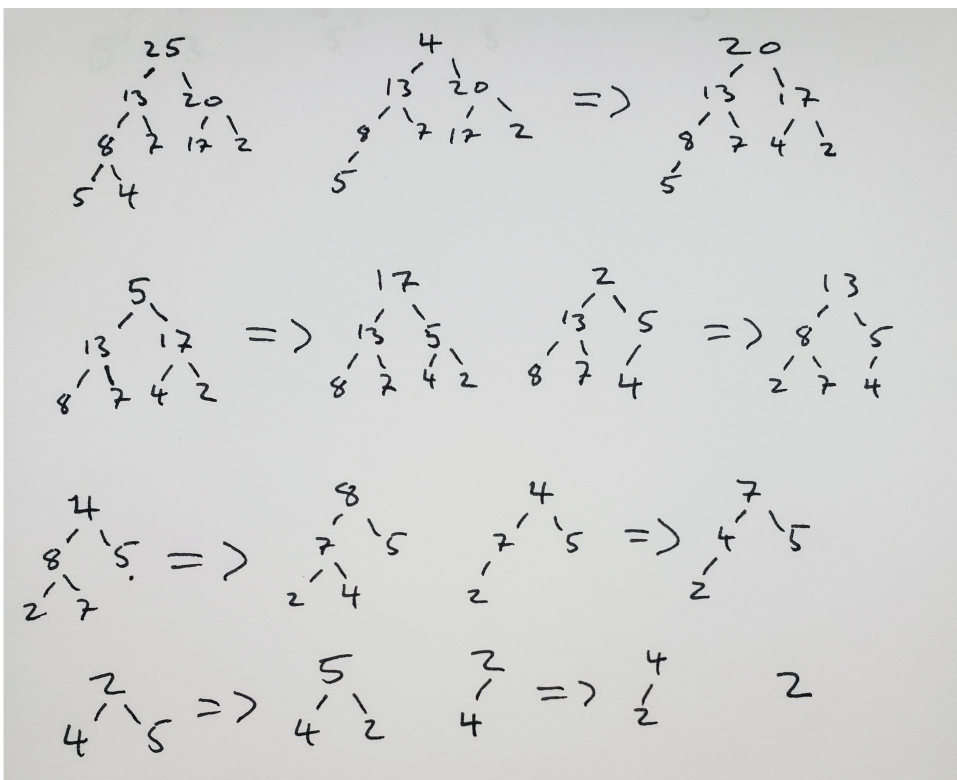
6(b).



7.

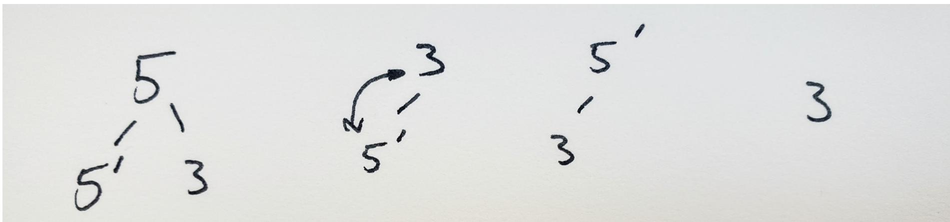
First, we create the max heap, and then we proceed with the deleting procedure.





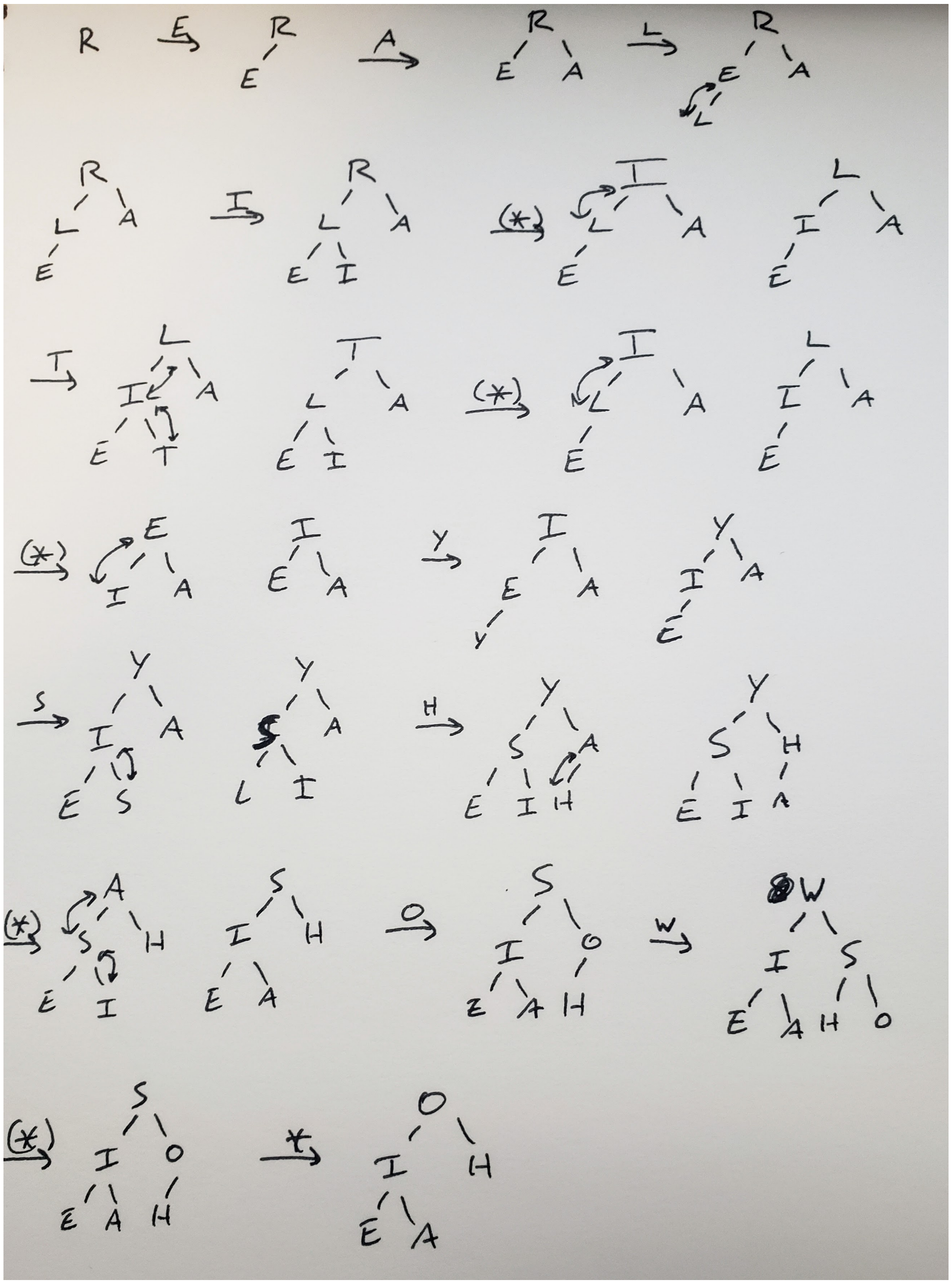
8.

A counterexample is if we want to sort the array 5, 5’, 3, where we denote the second 5 with 5’.



Then the sorted array becomes 3, 5, 5’. So the two fives are swapped.

9.



10. The frequency and distribution arrays are as follows.

Array values	a	b	c	d
Frequency	2	3	2	1
Distribution	2	5	7	8

Then we build the sorted array.

	$D[0\dots3]$				Sorted array							
A[7] = b	2	5	7	8					b			
A[6] = a	2	4	7	8		a						
A[5] = a	1	4	7	8	a							
A[4] = b	0	4	7	8				b				
A[3] = c	0	3	7	8							c	
A[2] = d	0	3	6	8								d
A[1] = c	0	3	6	7						c		
A[0] = b	0	3	5	7			b					

So the sorted array is a, a, b, b, b, c, c, d.