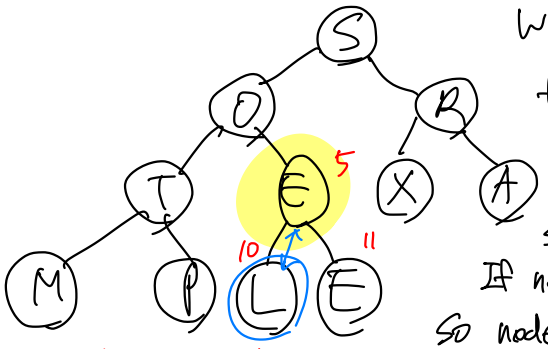
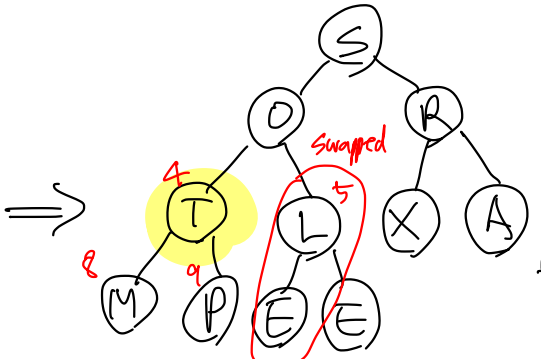
On my honor, I pledge that I have neither recieved nor provided any improper assistance in the completion of this assignment.
-21900112 김성민

Unsorted: | S | O | R | T | E | X | A | M | P | L | E |



K=5, N=11

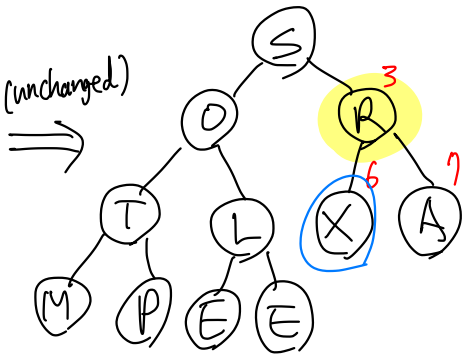
We have to start from node 5.
first, compare two child nodes (10 & 11.
L is larger than E, so node 10 should be considered.
If node 5 & 10 compared, L is larger than E.
So node 5 & 10 should be swapped.



K=4, N=11

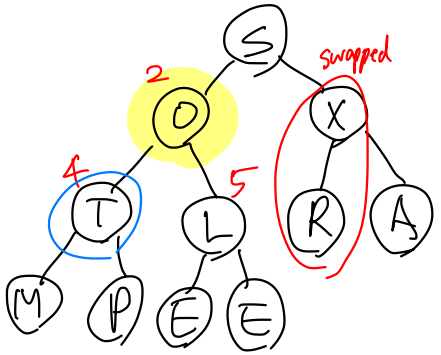
This is the result.
Since we checked node 5, we now have to check node 4 and its children.
First, compare node 8 & 9. P is larger than M, so we have to consider node 9.
If we compare node 4 & 9, T is larger than P. So, it is left unchanged.

(unchanged)



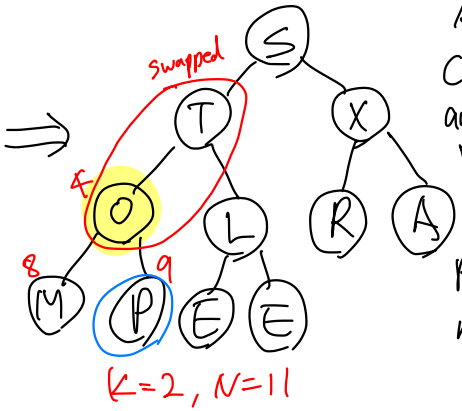
K=3, N=11

We now have to check node 3 after 5 & 4.
Larger child node among A & X is X.
Then, if X is compared to R, X is larger.
SO node 3 and 6 has to be swapped.



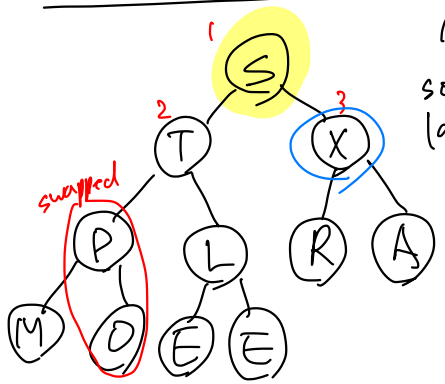
K=2, N=11

Checking node 2. T is larger than L, so we have to compare T & O (node 2 & 4). T is larger than O, so node 2 & 4 has to be swapped.



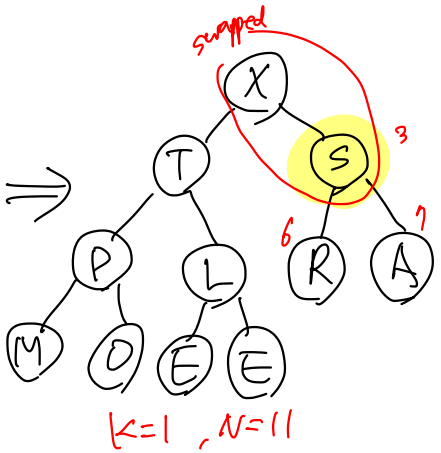
K=2, N=11

After swapping, because there are Children, we have to compare again and swap if necessary. And in this case, k is still 2. P is larger among M & P, we have to compare P & O. since P is larger, we need to swap node 4 & 9

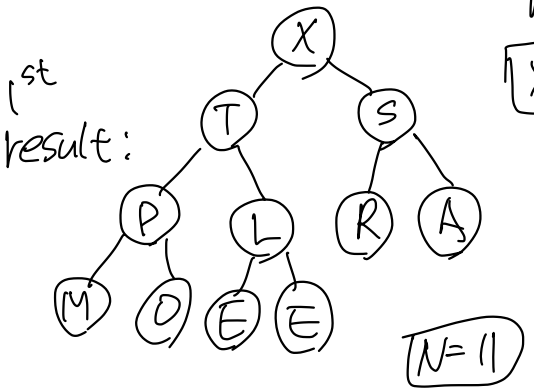


K=1, N=11

Now check node 1. X is larger than T, so we have to compare X & S. Since X is larger than T, we need to swap node 1 & 3. And node 3 has children, we have to compare again for additional swap.



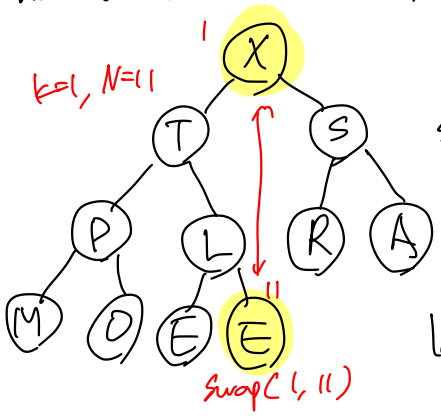
for node 3, consider largest child.
R is larger than A, so we have to compare S & R. Since S is larger, no additional swap is needed.
k value is still 1.



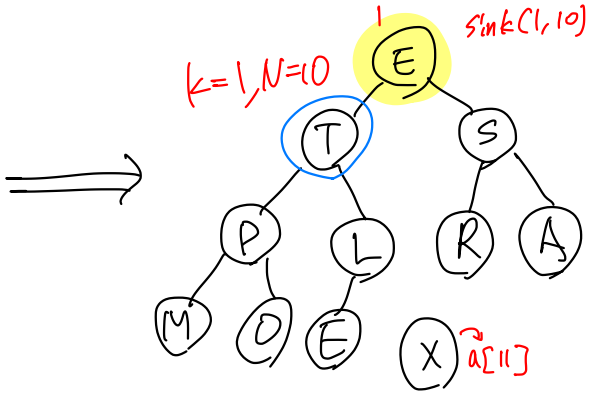
heap ordered array

1	2	3	4	5	6	7	8	9	10	11
X	T	S	P	L	R	A	M	O	E	E

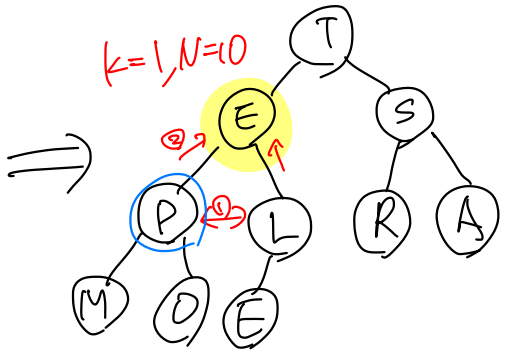
now we have max-heap ordered array, we can get sorted result by repeatedly removing maximum node, which is root node at every sequence.

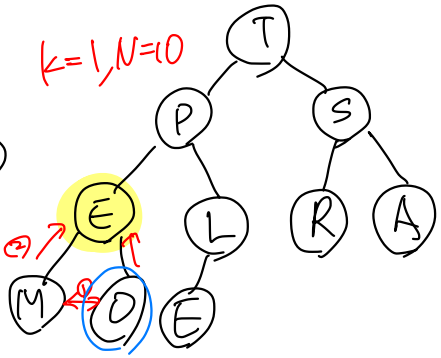


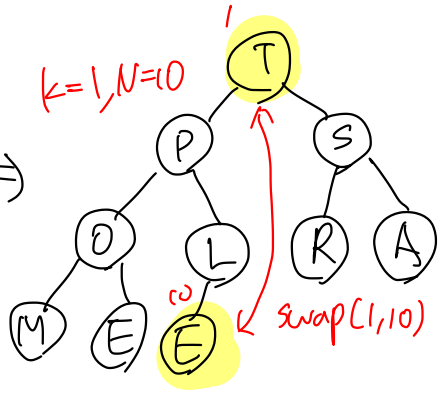
from heap-sorted, first swap node 1 & N, which is swap(1,11) in this case.
And then call sink(1, N-1).
Decrement has to be done because largest has to be removed.

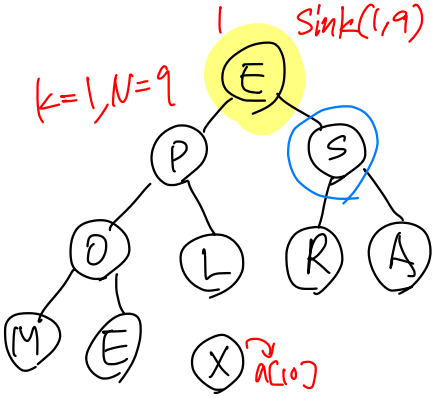


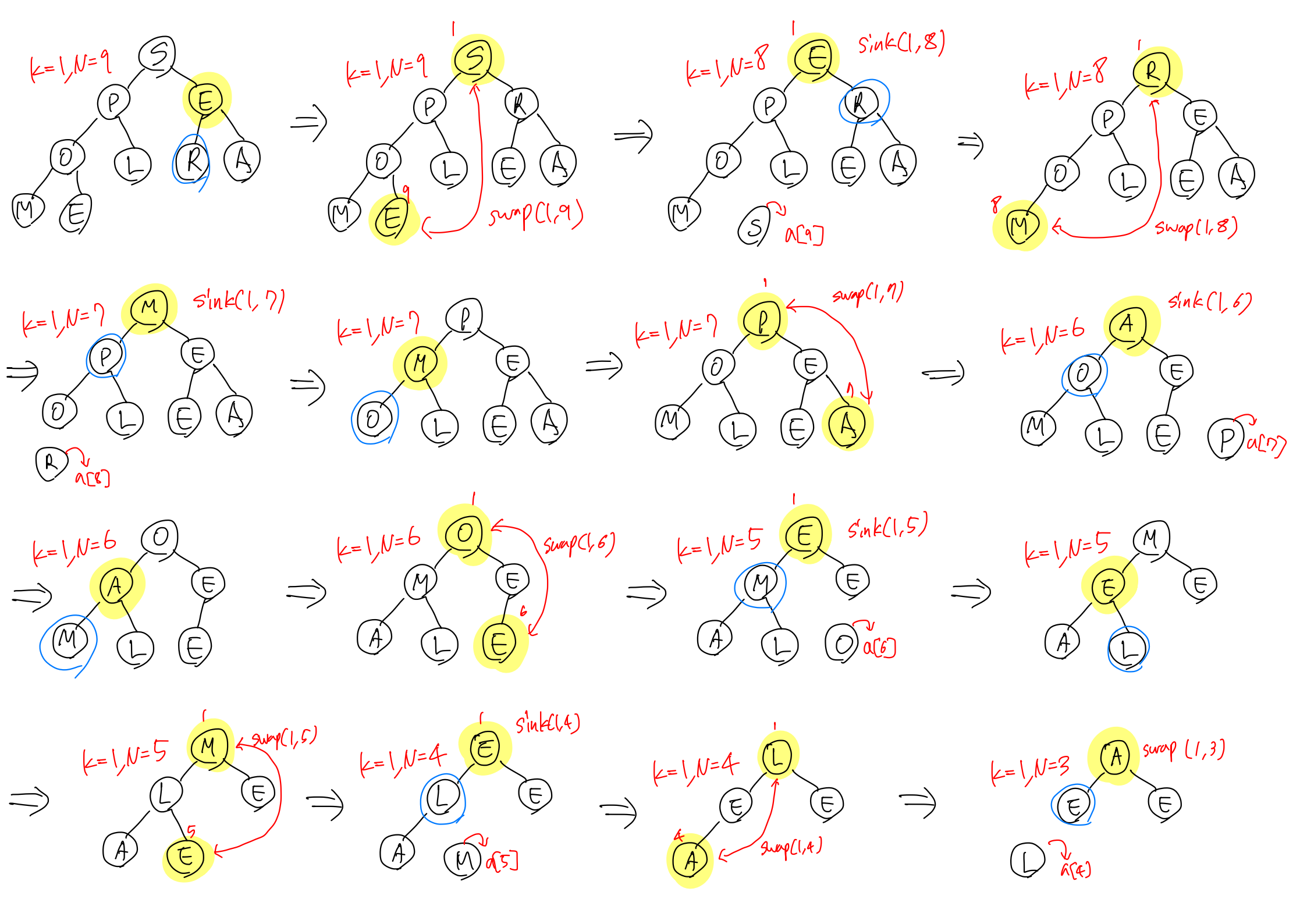
Now we have to find proper place for E.
So we have to invoke sink(1, 10). by sink, it will compare children and find its right position. As a explained algorithm, I will just show by diagram from now on.

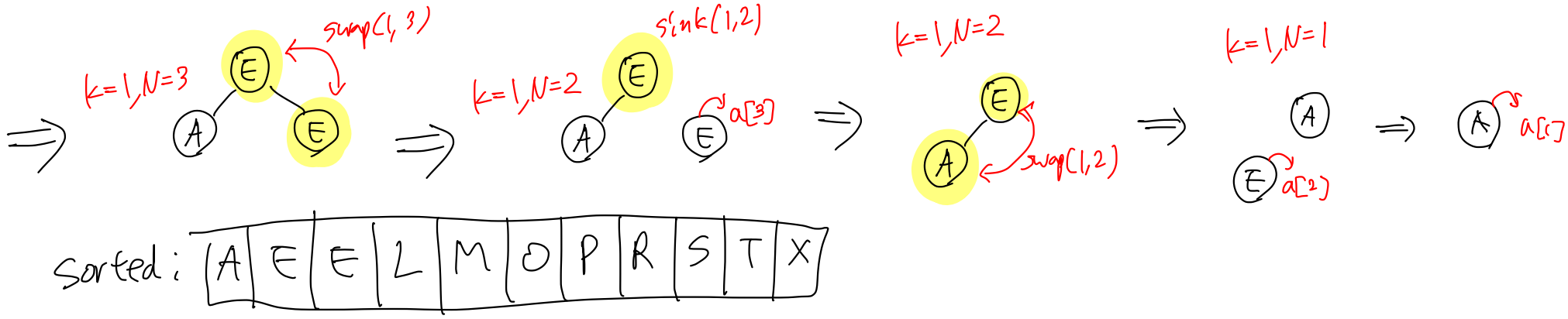




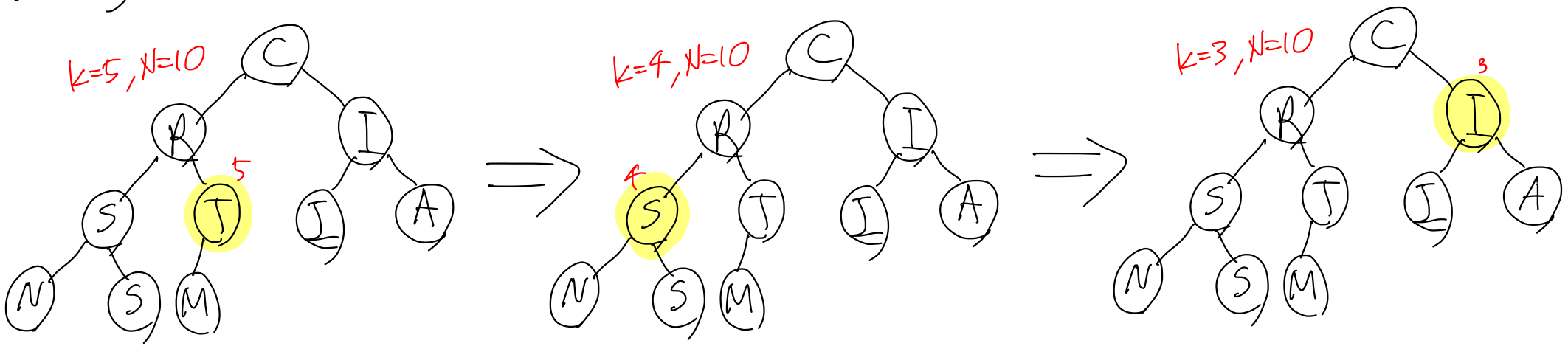


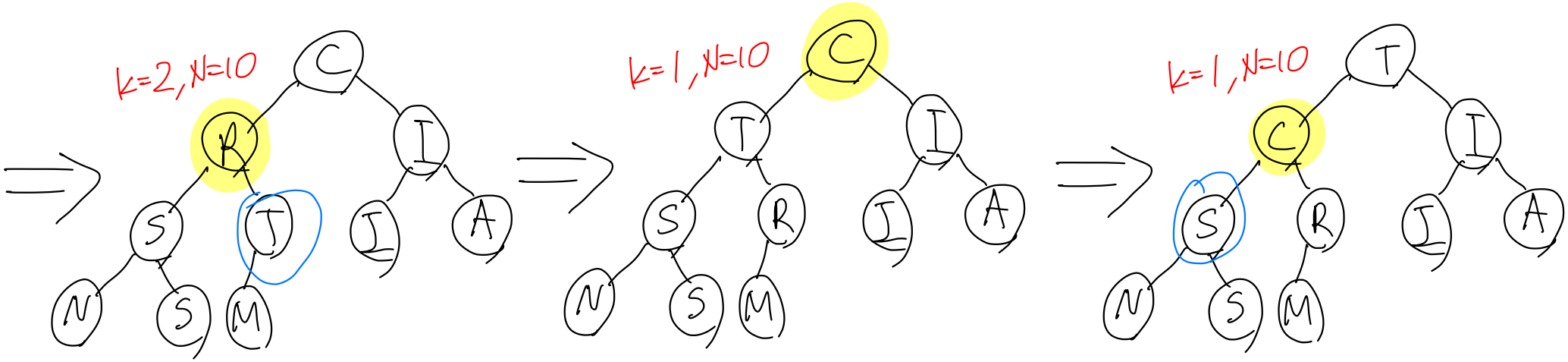


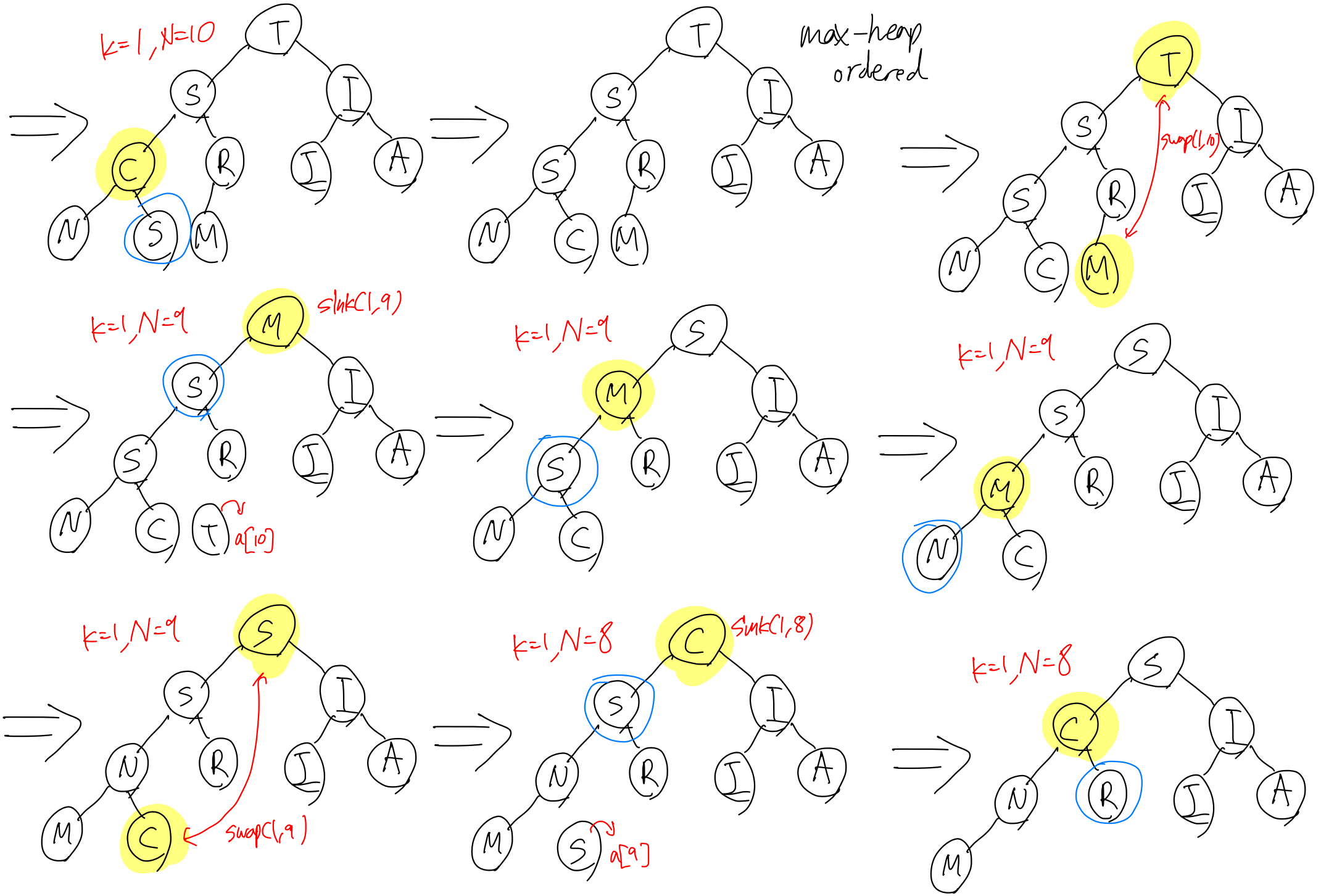

k=1,N=9
k=1,N=9
swap(1,9)
k=1,N=8
sink(1,8)
a[9]
k=1,N=8
swap(1,8)
k=1,N=7
sink(1,7)
a[8]
k=1,N=7
k=1,N=7
swap(1,7)
k=1,N=6
sink(1,6)
a[7]
k=1,N=6
k=1,N=6
swap(1,6)
k=1,N=5
sink(1,5)
a[6]
k=1,N=5
k=1,N=5
swap(1,5)
k=1,N=4
sink(1,4)
a[5]
k=1,N=4
swap(1,4)
k=1,N=3
swap(1,3)
a[4]

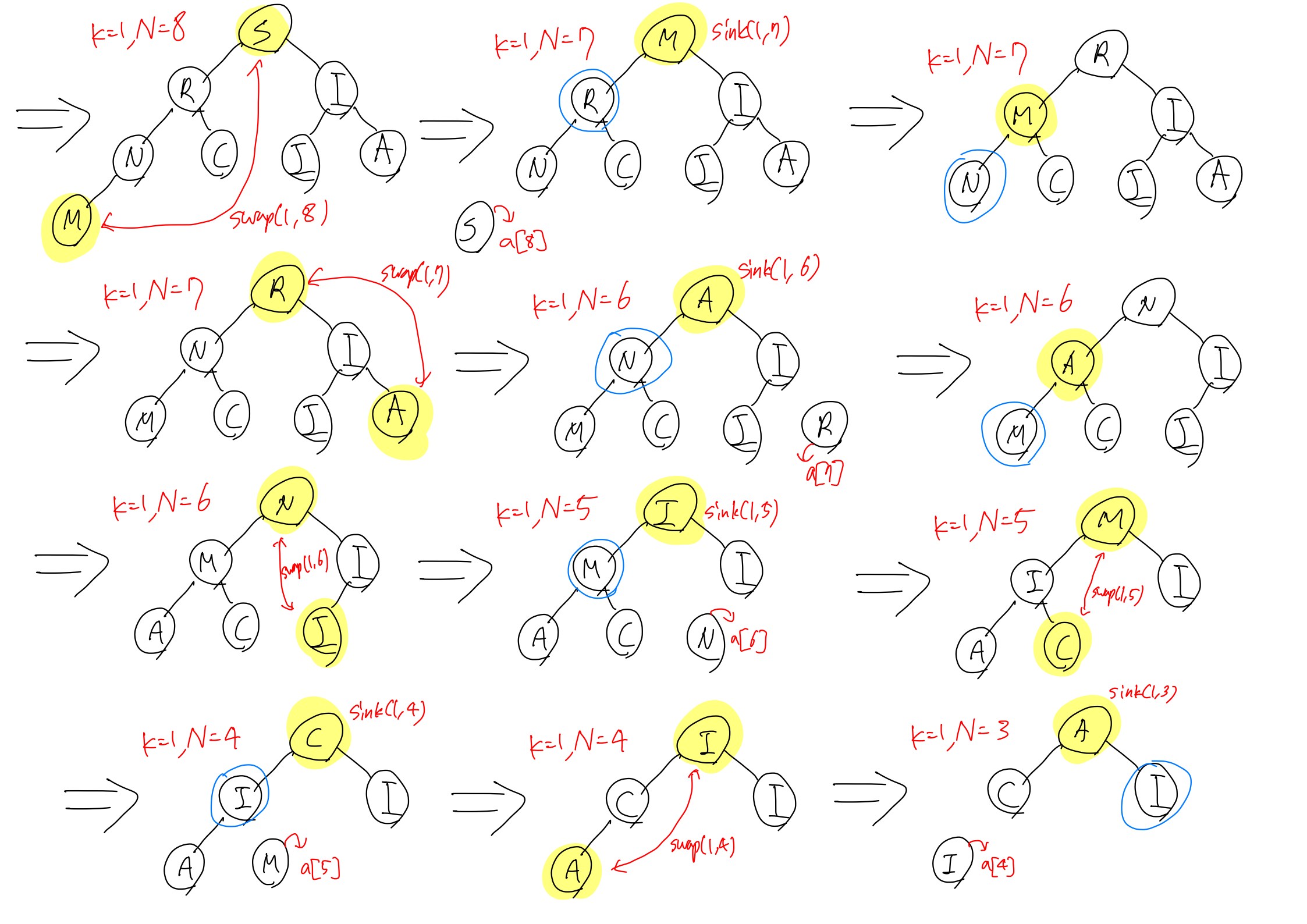


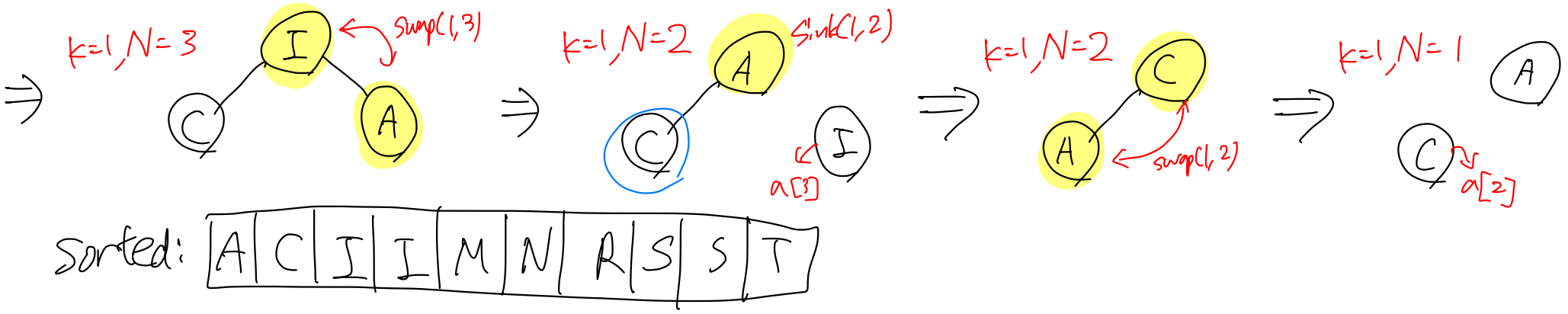
Sorting "CRISTIANSM"






k=1, N=10
max-heap ordered
swap(1,10)
k=1, N=9
sink(1,9)
a[10]
k=1, N=9
k=1, N=9
k=1, N=9
swap(1,9)
k=1, N=8
sink(1,8)
a[9]
k=1, N=8


k=1, N=8
swap(1,8)
k=1, N=7
sink(1,7)
S
a[8]
k=1, N=7
k=1, N=7
swap(1,7)
k=1, N=6
sink(1,6)
R
a[7]
k=1, N=6
k=1, N=6
swap(1,6)
k=1, N=5
sink(1,5)
N
a[6]
k=1, N=5
swap(1,5)
k=1, N=4
sink(1,4)
M
a[5]
k=1, N=4
swap(1,4)
k=1, N=3
sink(1,3)
I
a[4]



sorted: A C I I M N R S S T

Q. A) 1st pass : "S" → 4 comparisons, "M" → 2 comparisons

B) 2nd pass : "S" → 4 comparisons, "M" → 9 comparisons

```
→ pset13-14 git:(main) × ./heapsortx
thequickbrownfoxjumpsoveralazydog.
Enter a word to sort: SORTEXAMPLE
Input String:[ SORTEXAMPLE ], N=11
Input  a[11]: S O R T E X A M P L E

ASCENDING:
1st pass(heapify - O(n)) begins:
   N=11 k=5 S O R T L X A M P E E
   N=11 k=4 S O R T L X A M P E E
   N=11 k=3 S O X T L R A M P E E
   N=11 k=2 S T X P L R A M O E E
   N=11 k=1 X T S P L R A M O E E
HeapOrdered: X T S P L R A M O E E
2nd pass(swap and sink - O(n log n) begins:
   N=10 k=1 T P S O L R A M E E
   N=9 k=1 S P R O L E A M E
   N=8 k=1 R P E O L E A M
   N=7 k=1 P O E M L E A
   N=6 k=1 O M E A L E
   N=5 k=1 M L E A E
   N=4 k=1 L E E A
   N=3 k=1 E A E
   N=2 k=1 E A
   N=1 k=1 A
a[11]: A E E L M O P R S T X
```

```
→ pset13-14 git:(main) × ./heapsortx
thequickbrownfoxjumpsoveralazydog.
Enter a word to sort: CRISTIANSM
Input String:[ CRISTIANSM ], N=10
Input  a[10]: C R I S T I A N S M

ASCENDING:
1st pass(heapify - O(n)) begins:
   N=10 k=5 C R I S T I A N S M
   N=10 k=4 C R I S T I A N S M
   N=10 k=3 C R I S T I A N S M
   N=10 k=2 C T I S R I A N S M
   N=10 k=1 T S I S R I A N C M
HeapOrdered: T S I S R I A N C M
2nd pass(swap and sink - O(n log n) begins:
   N=9 k=1 S S I N R I A M C
   N=8 k=1 S R I N C I A M
   N=7 k=1 R N I M C I A
   N=6 k=1 N M I A C I
   N=5 k=1 M I I A C
   N=4 k=1 I C I A
   N=3 k=1 I C A
   N=2 k=1 C A
   N=1 k=1 A
a[10]: A C I I M N R S S T
```