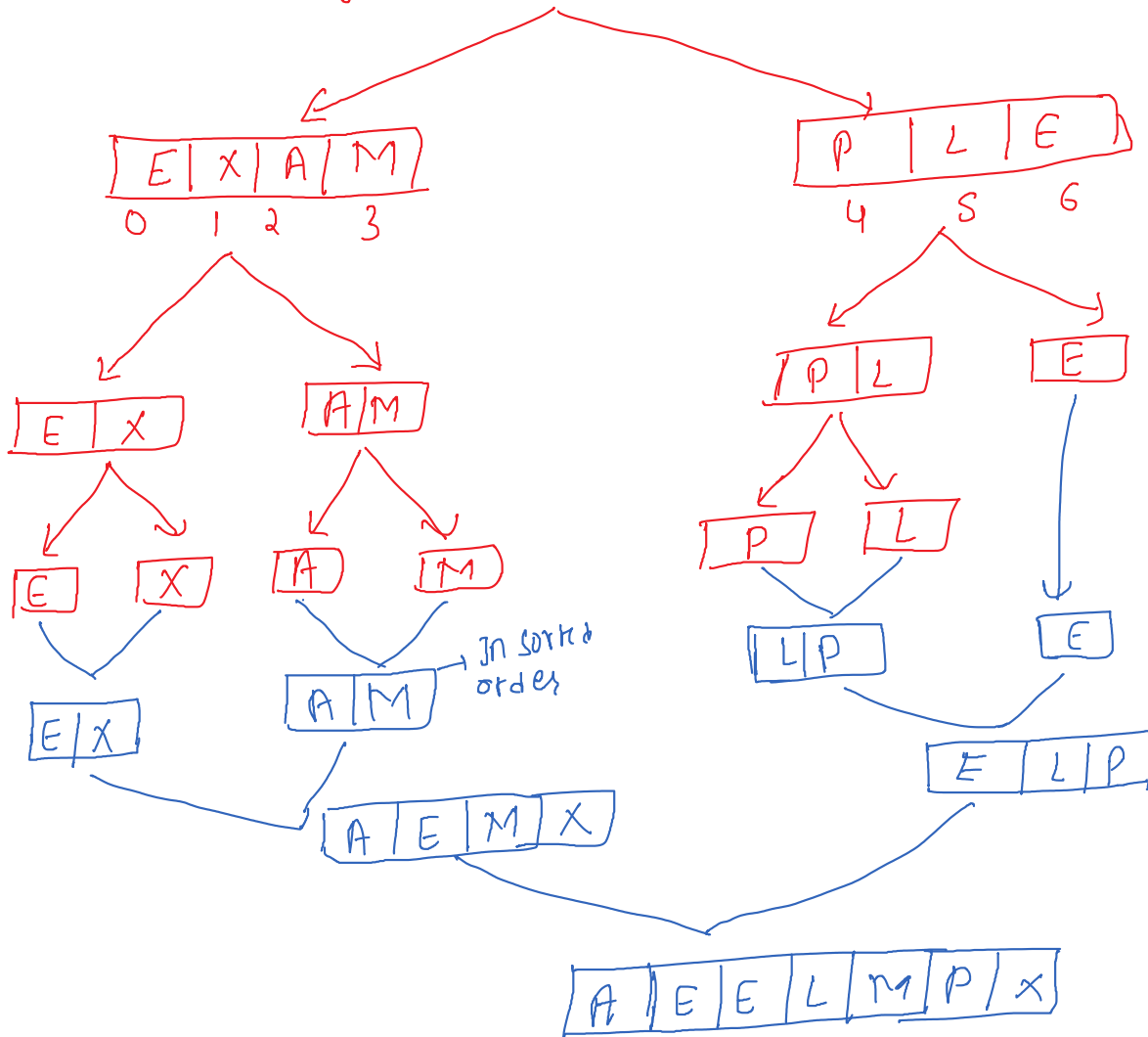


Ans (B)

lb (lower bound) ub (upper bound)
 APPLY merge Sort
 (subdivided the array)



* Using Algorithm :-

mergesort (A, lb, ub)

{ if (lb < ub)

{ mid = (lb + ub) / 2

mergesort (A, lb, mid);
 mergesort (A, mid+1, ub);
 merge (A, lb, mid, ub);

// Recursive call

}

mid = 4 (we take)

Recursive call

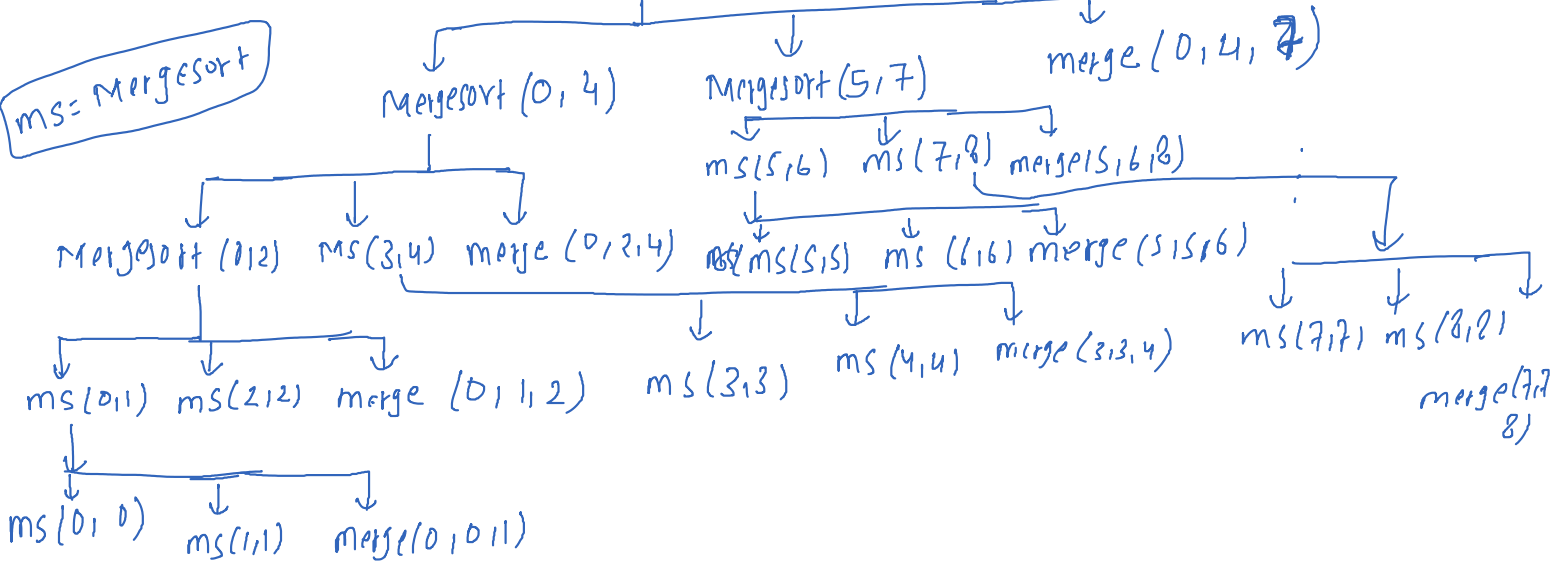
Recursive call

3

mid = 4 (we take)

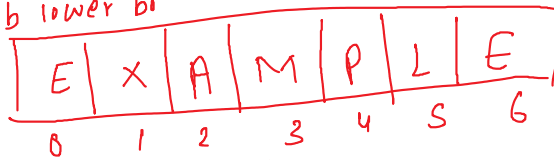
Mergesort (0, 7)

ms = Mergesort



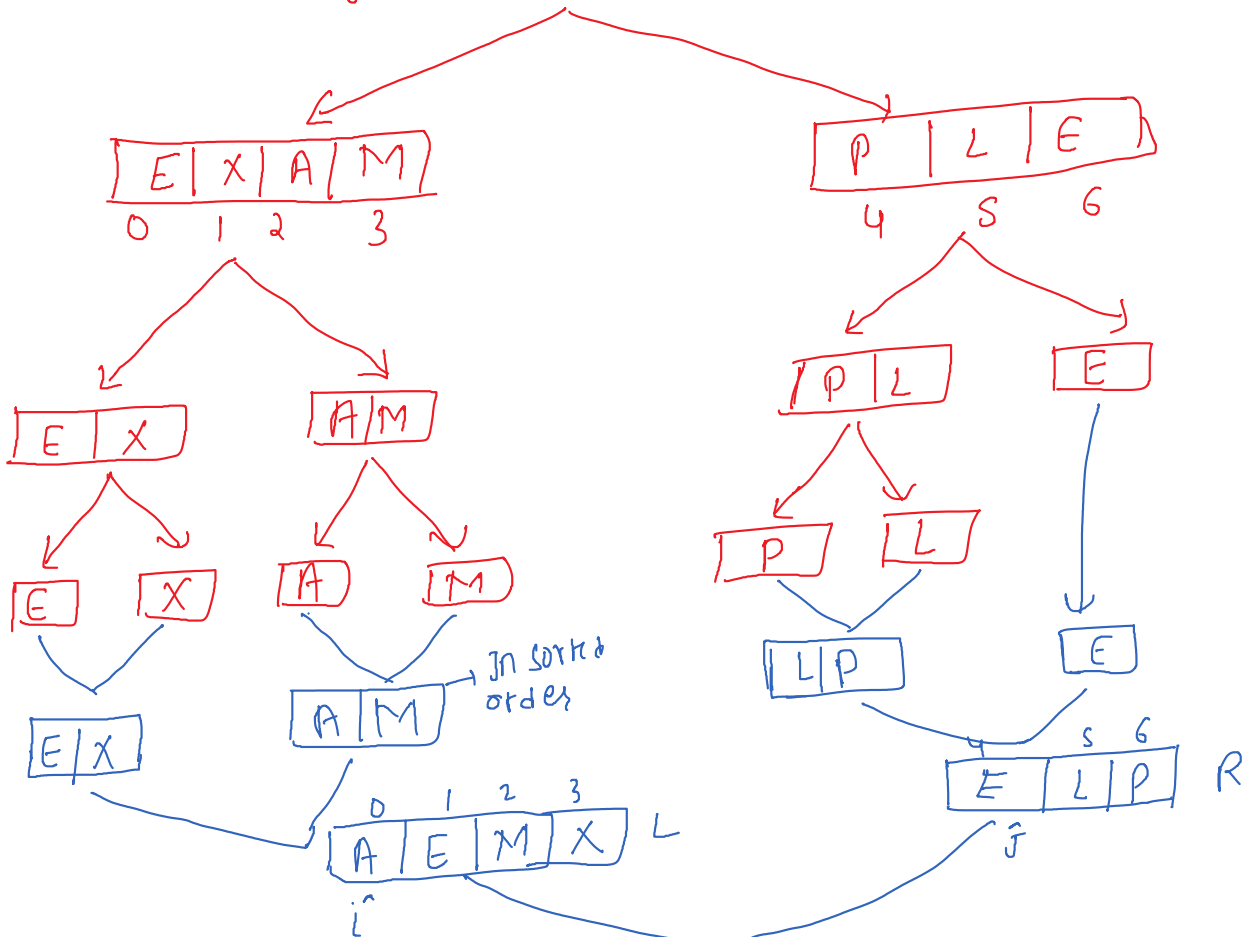
(B)

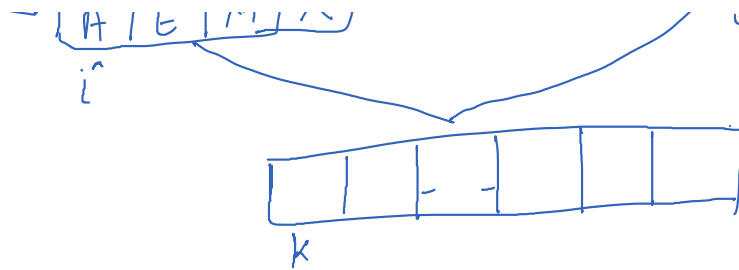
1b lower bi



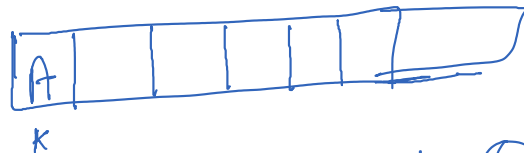
Apply m

(subdivided the a)





Step 01: $\text{if } (L(i) < R(j))$
 $k++;$



02: compare again k increment by ①



* Merge (A, lb, mid, ub)

{ $i = lb;$
 $j = mid + 1;$
 $k = lb;$

while ($i \leq mid$ && $j \leq ub$)

{ $\text{if } (a[i] \leq a[j])$

{ $b[k] = a[i];$

$i++;$

}

else {

$b[k] = a[j];$

$j++;$

} $k++;$

$\text{if } (i \leq mid)$

{ while ($j \leq ub$)

{ $b[k] = a[j];$

```

j++;
k++;
}
else { while (i2 = mid)
        { b[k] = a[i]
          i++;
          k++;
        }

```