

- 1) (i) class: QQQ
base class: PPP
derived class: RRR
- (ii) disp() can access M (because disp() is a member function of RRR), U (because RRR is a subclass of QQQ and can access protected variables in superclass) thus the same is true of S (for RRR is a subclass of PPP). Unless disp() called any functions in RRR it cannot access T or unless it calls member functions of PPP it cannot access H. disp() can only access M, U, S
- (iii) disp(), outdata(), indata(int, int), input(), out()
- (iv) yes

3. Pseudo Code written using www.geeksforgeeks.org/merge-sort/

```
(a) merge (array, mid, left, right)
{
    array_size-1 = mid - left + 1
    array_size-2 = right - mid

    a1 = left half of array
    a2 = right half of array

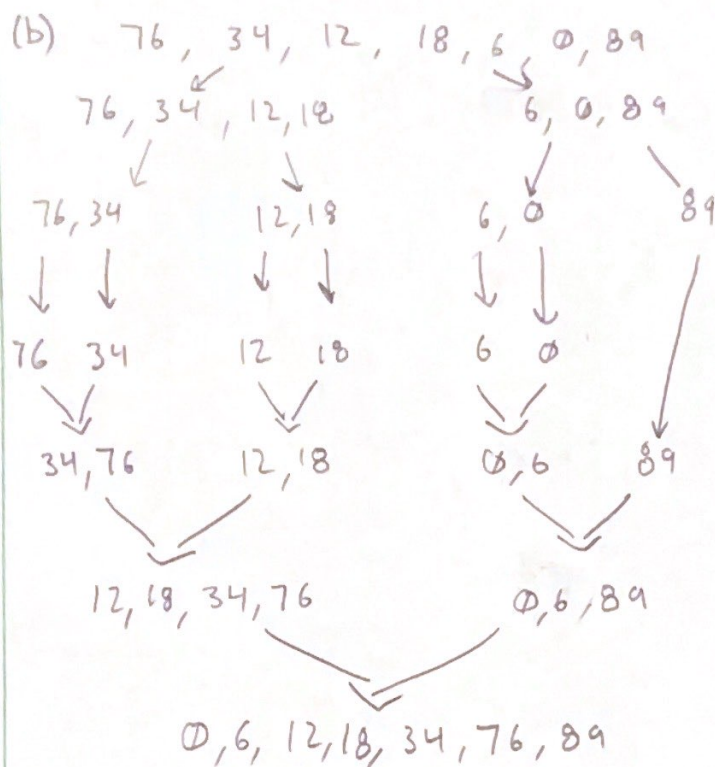
    while index1 < array_size1 && index2 < array_size2
        if left_val < right_val
            merged-a ← left_val
        else
            merged-a ← right_val

    if values left in one array but not other
        merged-a ← remaining values
}

mergeSort (array, start, end)
{
    if start >= end
        return // base case

    mid = middle of start and end
    mergeSort (array, start, mid)
    mergeSort (array, mid, end)
    merge (both arrays)
```

3. (cont.)



(c) array of n terms can be divided $\frac{n}{2}$ times
and sorted through iteratively n times
 $O(n) = 2O(n/2) + n$

if you keep dividing array k times you get

$$O(n) = 2^k O(n/2^k) + kn$$

$$n = 2^k \quad k = \log N$$

$$O(N) = NO(1) + N \log N = O(N \log N)$$

proof utilized from:

iq.opengenus.org/time-complexity-of-merge-sort