1.

```
 /** Merges array
                        * @param a first array
                        * @param b second array
                        * @param lastA number of "real" elements in a
                        * @param lastB number of "real" elements in b
                        */
                        public static void merge(int[] a, int[] b, int lastA, int lastB) {
                        int indexMerged = lastB + lastA - 1; /* Index of last location of merged array */
                        int indexA = lastA - 1; /* Index of last element in array b */
                        int indexB = lastB - 1; /* Index of last element in array a */
                        /* Merge A and B, start from the end of the array.*/
                        while(indexB>=0)
                        {
                        if(indexA>=0 && a[indexA] >= b[indexB])
                        {
                        a[indexMerged] = a[indexA];
                        indexA--;
                        }
                        else
                        {
                        a[indexMerged] = b[indexB];
                        indexB--;
                        }
                        indexMerged--;
                        }

                        }
```

Yes, merge in reverse order to the back of the larger array.

2. Flase

3. Depends on the class (monday class started on 0, wednesday class started on 1):

1,0,2,3,4,5

0,1,2,3,5,4

4. The answer should mention that all nodes will be included in the minimum spanning tree. ie prim is finding a mst.

5. A,E,F,B,D,G (there may be some other answers)

6.

a) should be a graph with 4 nodes.

A connects an arrow to B, C, D

B arrow to A

C arrow to B and D

b)

a 4 X 4 matrix

F T T T

T F F F

F T F T

F F F F

7.

a)

mentioned some nested arrays to search for duplicates,

b) mentioned using some merge sort or quicksort (any nlogn) sort which also includes searching for duplicates. Must have both search + duplicate searching technique described to get full points

8. Note this question was asking about space efficiency, not runtime efficiency. So I wanted to see graphs which have short path length overall from the start node, but at each degree there are several nodes.

9. No, and proved by counter example .. some graph with a negative degree edge which causes a cycle for Dijkstra, or any kind of graph which you showed Dijkstra resulted in the wrong answer.