

Review

1. What is the output of the following algorithm when $n = 6$, $n = 8$, and $n = 10$?
2. What is the time complexity $T(n)$? You may assume n is divisible by 2.

```
j = 1;
while ( j <= n / 2)
    i = 1;
    while ( i <= j)
        cout << j << i;
        i++;
    j++;
```

Review

3. Draw the recursion tree that is created when `Quicksort` and `Mergesort` are performed on this array. Label the ordering of steps for `MergeSort`.
{ 2, 4, 11, 5, 3, 8, 10, 1, 3, 7, 15, 12 }
4. Draw a chart reflecting a top-level call to `partition` on the same array.

Review

5. Write a function `doubleArray` that takes an integer array of any size as a parameter. This function should double the size of the array

Review

6. Write a recursive algorithm that searches a sorted list of n items by dividing it into three sublists of almost $n/3$ items each. This algorithm will test the element at position $n/3$ and the element at position $2n/3$. It finds the sublist that might contain the target item, and divides the list into three smaller sublists of almost equal size. It repeats this process until it finds the item or concludes that it is not in the list.
7. Analyze your algorithm by defining its recurrence relation, and give the worst-case time complexity result.