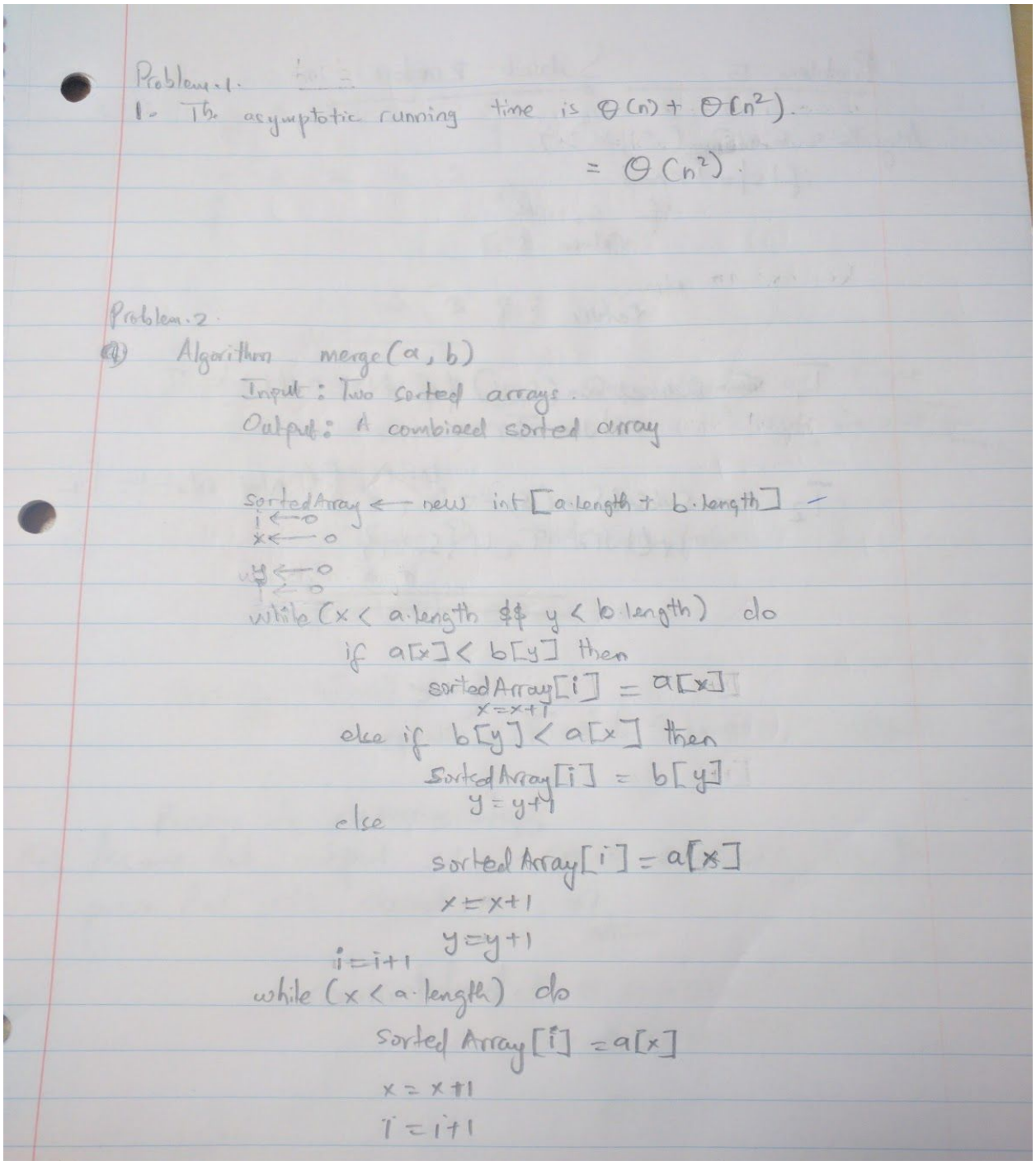
Problem 1 & Problem 2



Problem 1.

1. The asymptotic running time is $\Theta(n) + \Theta(n^2)$.

$= \Theta(n^2)$.

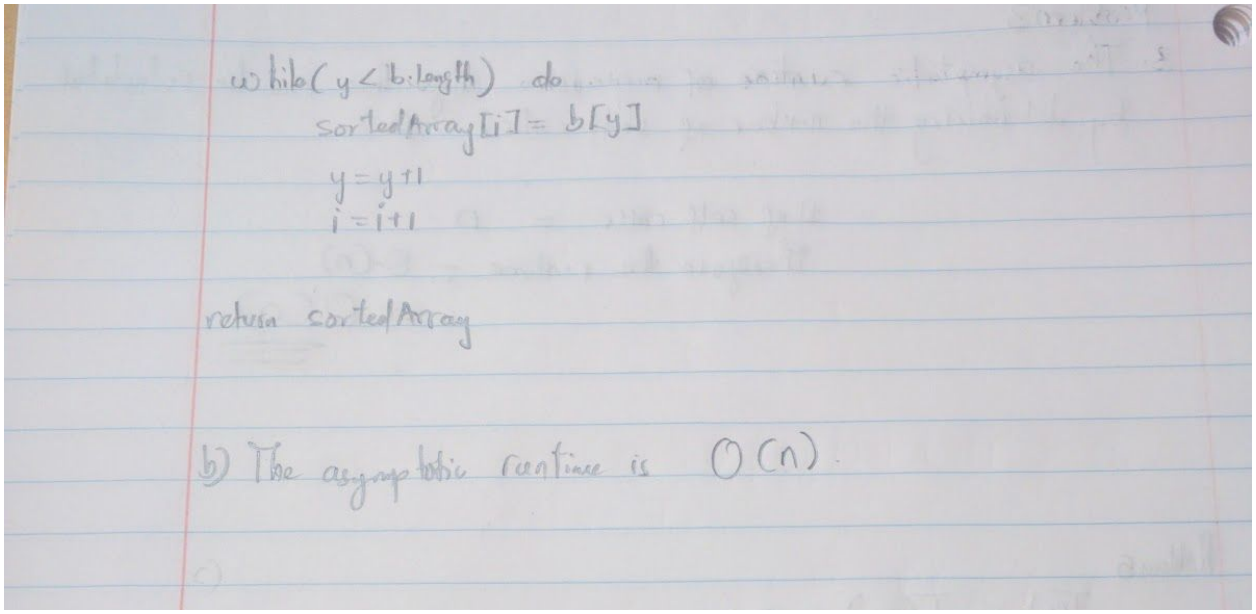
Problem 2.

(a) Algorithm merge(a, b)

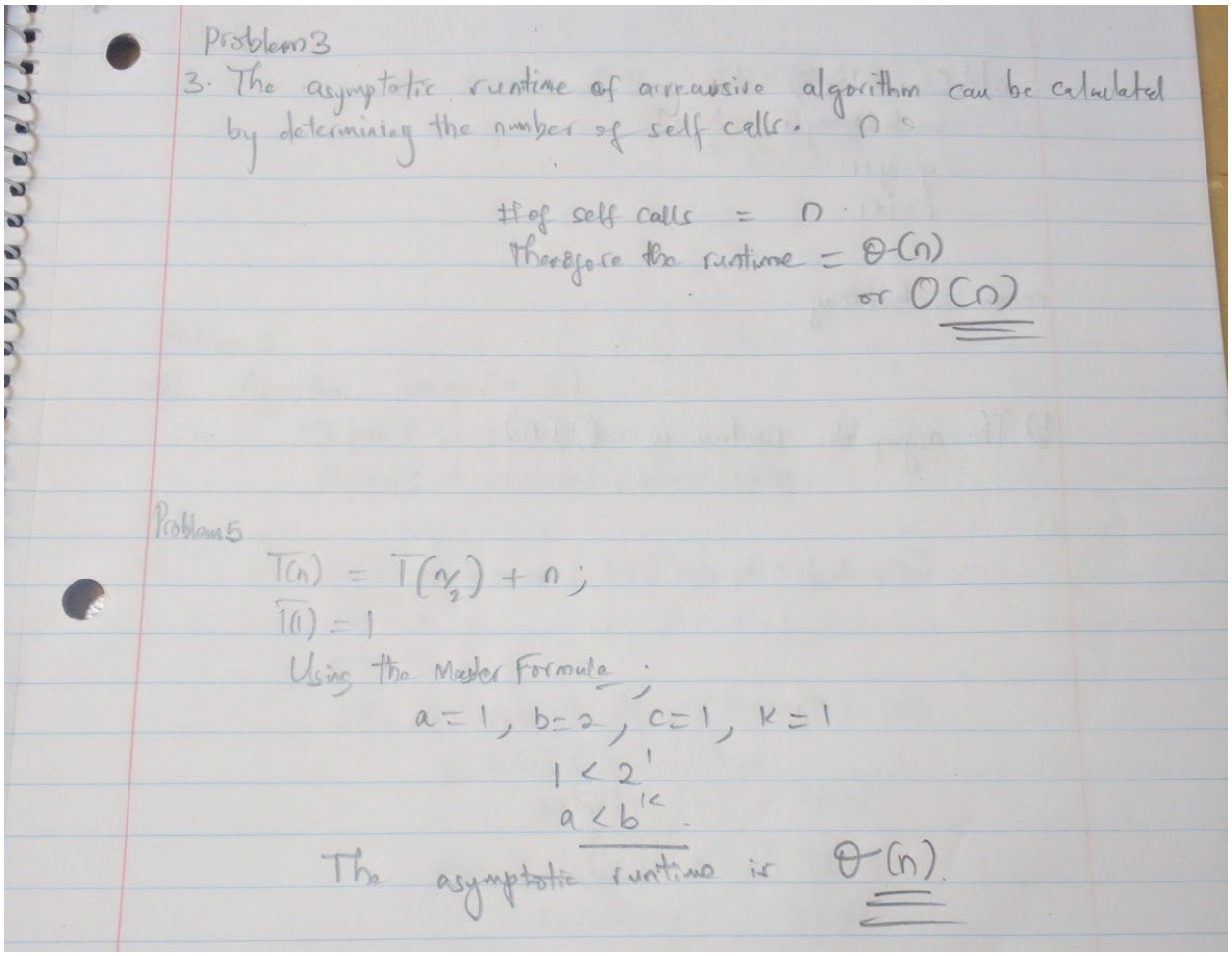
Input : Two sorted arrays.

Output: A combined sorted array

```
sortedArray ← new int[a.length + b.length]
i ← 0
x ← 0
y ← 0
while (x < a.length && y < b.length) do
    if a[x] < b[y] then
        sortedArray[i] = a[x]
        x = x+1
    else if b[y] < a[x] then
        sortedArray[i] = b[y]
        y = y+1
    else
        sortedArray[i] = a[x]
        x = x+1
        y = y+1
    i = i+1
while (x < a.length) do
    sortedArray[i] = a[x]
    x = x+1
    i = i+1
```



Problem 3 & Problem 5



Problem 4

```
public LinkedList<Set<Integer>> powerSet(LinkedList<Integer>
inputList){
      LinkedList<Set<Integer>> resultSet = new LinkedList<>();
      Set<Integer> emptySet = new HashSet<>();
      resultSet.add(emptySet);
      while(!inputList.isEmpty()){
            Integer f = inputList.removeFirst();
            int size = resultSet.size();
            //Set<Integer> currentSet;
            for(int i = 0; i < size; i++){
                  Set<Integer> union = new HashSet<>();
                  Set<Integer> currentSet = resultSet.get(i);
                  union.add(f);
                  union.addAll(currentSet);
                  resultSet.add(union);
            }
      }
      return resultSet;
}
```