Maximum Tables Occupied

Your friend has opened a new café. You wanted to know how occupied his café was yesterday. Instead of giving a direct answer, he let you know the arrival and departure times of different groups at his café [This is your input]. Each group occupies one table. You must find the maximum number of tables that were occupied at a time there.

[Hint]:
1. Sort the arrival array
2. Sort the departure array
3. Try to merge the sorted arrays
4. Keep track of the number of groups inside the café during each iteration
5. And also keep track of the maximum number at any time

Sample Input	Sample Output
arrival = {2.00, 2.10, 3.00, 3.20, 3.50, 5.00} departure = {2.30, 3.40, 3.20, 4.30, 4.00, 5.20}	2
arrival = {9.00, 9.40, 9.50, 11.00, 15.00, 18.00} departure = {9.10, 12.00, 11.20, 11.30, 19.00, 20.00}	3

Explanation for the first sample input:
1. Sort the arrival array
   a. arrival = {2.00, 2.10, 3.00, 3.20, 3.50, 5.00}
2. Sort the departure array
   a. departure = {2.30, 3.20, 3.40, 4.00, 4.30, 5.20}
3. Merge the two sorted arrays into a new array so that the merged array is also sorted. Also, keep track of which time is arrival and which is departure.
   a. Merged = {(2.00,a), (2.10,a), **(2.30,d)**, (3.00,a), (3.20,a), **(3.20,d)**, **(3.40,d)**, (3.50,a), **(4.00,d)**, **(4.30,d)**, (5.00,a), **(5.20,d)**} // a= arrival, d=departure
4. Keep track of the number of tables occupied at the café during each iteration
       #tables occupied
   - (2.00,a) -> 1
   - (2.10,a) -> 2
   - **(2.30,d)** -> 1
   - (3.00,a) -> 2
   - (3.20,a) -> 3
   - **(3.20,d)** -> 2
   - **(3.40,d)** -> 1
   - (3.50,a) **-> 2**
   - **(4.00,d)** -> 1

- **(4.30,d)** -> 0
- (5.00,a) -> 1
- **(5.20,d)** -> 0

5. **Answer 2**; you may think it is 3 but it is 2 because at 3.20 even though 1 group arrives, one leaves immediately at 3.20 too