# Coding challenge :: Function :: II

Sometimes when a customer is charged, there is a duplicate transaction created. We need to find those transactions so that they can be dealt with. Everything about the transaction should be identical, except the transaction id and the time at which it occurred, as there can be up to a minute delay.

Create a function `findDuplicateTransactions(transactions)` to find all transactions that have the same `sourceAccount`, `targetAccount`, `category`, `amount`, and the time difference between each consecutive transaction is less than 1 minute.

### Input

You can assume that all parameters will always be present and valid. However, the incoming transactions are not guaranteed to be in any particular order.

### Output

List of all the duplicate transaction groups, ordered by time ascending.

### Example

Input transactions:

```
[
  {
    id: 3,
    sourceAccount: 'A',
    targetAccount: 'B',
    amount: 100,
    category: 'eating_out',
    time: '2018-03-02T10:34:30.000Z'
  },
  {
    id: 1,
    sourceAccount: 'A',
    targetAccount: 'B',
```

~~~
    amount: 100,
    category: 'eating_out',
    time: '2018-03-02T10:33:00.000Z'
  },
  {
    id: 6,
    sourceAccount: 'A',
    targetAccount: 'C',
    amount: 250,
    category: 'other',
    time: '2018-03-02T10:33:05.000Z'
  },
  {
    id: 4,
    sourceAccount: 'A',
    targetAccount: 'B',
    amount: 100,
    category: 'eating_out',
    time: '2018-03-02T10:36:00.000Z'
  },
  {
    id: 2,
    sourceAccount: 'A',
    targetAccount: 'B',
    amount: 100,
    category: 'eating_out',
    time: '2018-03-02T10:33:50.000Z'
  },
  {
    id: 5,
    sourceAccount: 'A',
    targetAccount: 'C',
    amount: 250,
    category: 'other',
    time: '2018-03-02T10:33:00.000Z'
  }
];
```
~~~

Expected output:

```
[
  [
    {
      id: 1,
      sourceAccount: "A",
      targetAccount: "B",
      amount: 100,
      category: "eating_out",
      time: "2018-03-02T10:33:00.000Z"
    },
    {
      id: 2,
      sourceAccount: "A",
      targetAccount: "B",
      amount: 100,
      category: "eating_out",
      time: "2018-03-02T10:33:50.000Z"
    },
    {
      id: 3,
      sourceAccount: "A",
      targetAccount: "B",
      amount: 100,
      category: "eating_out",
      time: "2018-03-02T10:34:30.000Z"
    }
  ],
  [
    {
      id: 5,
      sourceAccount: "A",
      targetAccount: "C",
      amount: 250,
      category: "other",
      time: "2018-03-02T10:33:00.000Z"
    },
```

~~~
    {
      id: 6,
      sourceAccount: "A",
      targetAccount: "C",
      amount: 250,
      category: "other",
      time: "2018-03-02T10:33:05.000Z"
    }
  ]
];
```
~~~