Hints for Palindromes with Friends

- Given a bunch of letters, how do we know that they can be arranged into a palindrome, or not?
  - **If** these letters can form a palindrome, we know that:
    - First letter of palindrome is identical to last letter
    - Second letter is identical to second-last letter
    - ...
    - What if the word length is even? What about if odd? Do we have anything *left over*?
    - If we add a pair of identical letters to this bunch, can we still form a palindrome?
    - What about removing a pair of identical letters?
  - What about if the bunch of letters can **never form** a palindrome?
- For now, assume we only want to produce even-length palindromes.
  - We are given the bunch of letters in some bag. In **every** possible way to form some even-length palindromes...
    - ...which letters are **necessary** to add?
    - Is it **sufficient** to add only those letters?
  - If two bags combined can form an even-length palindrome, what does that say about each of their **necessary** letters?
- Now, we focus only only **odd**-length palindromes.
  - We are given the bunch of letters in some bag.

    In the even-length case, we had **necessary** letters.
    - In contrast, for the odd-length case, do we need to add every single one of them? How many do we actually need?
    - Is it **sufficient** to add only those letters?
  - If two bags combined can form an odd-length palindrome, what does that say about each of them? (i.e. X's unpaired letters is Y's unpaired letters + 1 more letter.)
- Can we group up 'similar' groups of bags under some common key?
  - How should we define 'similar'?

    Given some letter bag X, if we can pair it with bag Y, then X can also be paired with every other bag in the same group as Y.
  - Given a group of bags with key k_1, which other groups (with key k_2) should we pair with? What's the relation between k_1 and k_2?
  - How many possible groups are there at most?

  - How should we store these groups and associated keys? What kind of operations do we do on them?
    - For each group... (We want to iterate over existing groups/keys)
    - Given this key k_1, for each compatible key k_2... (We want to go through all compatible keys)
    - Find the bags with that fall in the group with key k_2.
    - Compute number of pairs. (Remember to avoid double counting both (i, j) and (j, i), and skip self-pairs like (i, i)).

- Important note: HashMaps assume their key (inserted by put()) never gets modified. If you use a composite/mutable key, make sure the key object passed to put(), or obtained via iteration, is never changed.