1.4 Palindrome Permutation - Given a string write a function to check if it is a permutation of a palindrome.

Parameters → (String: palin) → Can I always trust it's a string?
Return → True if str is a palin permutation else false.

Simplest Problem
IN- "a" → "ab"
OUT- True
IN - "abc"
OUT- False
IN - "cbc"
OUT - ~~False~~ True
IN - "bcc"
OUT- True
IN - "abcd"
OUT - False
IN- "adda"
Out - True
IN- "ab b b b a"

Edge Cases
→ No chars
→ All spaces
→

Toolbox
- Sorting
- Hash map/big array

Solutions
- Hash map of ints to hold counts make sure if odd there is one 1
- Sorting then checking count

Complexity
O(n) - time
O(1) - space

O(nLog(n))

Patterns - ignore spaces
- if odd → all chars should have even count excep 1
- If even → all chars should have even count

Code

```
def palinPerm(str):              → str= str.replace(' ',")
   dic ={}                          if len(str) %2 ==0
   for elem str:                       odd= True
                                       not found==False
      dic[elem]= dic.get(elem, 0) + 1
   for val in dic.values():
      if(val %2 == 0 or(val %1 and odd and notFound):
   else:
      return False

return True
```