1.5 One Away → There are three types of edits: insert, replace, delete. Determine if a ~~character~~ string is one char away from another (or zero).

Parameters → (String: str1, String: str2)
Return → True if <=1 operations away

Solution
- if lengths same → replace
if different check remove of larger.

Complexity
O(n) O(1) space

Simplest problem:
IN → "a" "", "a" "b", "" "a", "a", "a"
OUT → True
IN → "ab" "d"
Out → False
IN → "ac" "ab"

Edge Cases

Patterns
- Adding is same as removing from other string
- String length must be between -1 <= len(ot) <= 1

"pale" pale
"ple" pal

Code

```
def oneAway(str1, str2):
    if (len(str1) - len(str2)) > 1:
        return False
    largest = str1, smallest = str2
    if (len(str) == len(str2)):
        isOne = True
        for elem, elem2 in (str, str2):
            if elem != elem2 and isOne:
                isOne = False
            elif elem != elem2
                return isOne
    if (len(str1) < len(str2)):
        largest = str2
```

Math.abs()

```
i=0, j=0
while i < len(largest) and j < len(smallest):
    if largest[i] != smallest[j]:
        isfound = false
        if largest[i+1] == smallest[j]:
            i += 2
            j += 1
        else return false
    i += 1
    j += 1
return True
```