Reverse Vowels of a String

Write a function that takes a string as input and reverse only the vowels of a string.

Example 1:
Given s = "hello", return "holle".

Example 2:
Given s = "leetcode", return "leotcede".

Solution 1

In the inner while loop, don't forget the condition "start less than end" while incrementing start and decrementing end. This is my friend's google phone interview question. Cheers! // update! May use a HashSet to reduce the look up time to O(1)

```
public class Solution {
public String reverseVowels(String s) {
    if(s == null || s.length()==0) return s;
    String vowels = "aeiouAEIOU";
    char[] chars = s.toCharArray();
    int start = 0;
    int end = s.length()-1;
    while(start<end){

        while(start<end && !vowels.contains(chars[start]+"")){
            start++;
        }

        while(start<end && !vowels.contains(chars[end]+"")){
            end--;
        }

        char temp = chars[start];
        chars[start] = chars[end];
        chars[end] = temp;

        start++;
        end--;
    }
    return new String(chars);
}
```

}

written by ninacc original link here

Solution 2

Ruby

```ruby
def reverse_vowels(s)
  vowels = s.scan(/[aeiou]/i)
  s.gsub(/[aeiou]/i) { vowels.pop }
end
```

Python

```python
def reverseVowels(self, s):
    vowels = re.findall('(?i)[aeiou]', s)
    return re.sub('(?i)[aeiou]', lambda m: vowels.pop(), s)
```

It's possible in one line, but I don't really like it:

```python
def reverseVowels(self, s):
    return re.sub('(?i)[aeiou]', lambda m, v=re.findall('(?i)[aeiou]', s): v.pop(
), s)
```

Another version, finding replacement vowels on the fly instead of collecting all in advance:

```python
def reverseVowels(self, s):
    vowels = (c for c in reversed(s) if c in 'aeiouAEIOU')
    return re.sub('(?i)[aeiou]', lambda m: next(vowels), s)
```

written by StefanPochmann original link here

Solution 3

```
class Solution {
public:
    string reverseVowels(string s) {
        auto p1 = s.begin(), p2 = s.end() - 1;
        string vowels = "aeiouAEIOU";
        while(p1 < p2) {
            while((vowels.find(*p1) == string::npos) && (p1 < p2)) p1++;
            while((vowels.find(*p2) == string::npos) && (p1 < p2)) p2--;
            if(p1 < p2) swap(*p1, *p2);
            p1++;
            p2--;
        }
        return s;
    }
};
```

written by linjian2015 original link here