

给定 **S** 和 **T** 两个字符串，当它们分别被输入到空白的文本编辑器后，判断二者是否相等，并返回结果。 **#** 代表退格字符。

示例 1:

输入: S = "ab#c", T = "ad#c"

输出: true

解释: S 和 T 都会变成 "ac"。

示例 2:

输入: S = "ab##", T = "c#d#"

输出: true

解释: S 和 T 都会变成 ""。

示例 3:

输入: S = "a##c", T = "#a#c"

输出: true

解释: S 和 T 都会变成 "c"。

示例 4:

输入: S = "a#c", T = "b"

输出: false

解释: S 会变成 "c"，但 T 仍然是 "b"。

提示:

1. `1 <= S.length <= 200`
2. `1 <= T.length <= 200`
3. **S** 和 **T** 只含有小写字母以及字符 **#**。

思路，对于字符串的操作个人还是不太会，直接用vector来记录char数组，#就退一格，其余的新增，然后比较数组长度，不等返回false，相等就继续比较字符是否相同。需注意添加进vector数组时，若是vector长度等于0，则不进行出队列操作。

代码

```
class Solution {
public:
    bool backspaceCompare(string S, string T) {
        vector<char> s;
        vector<char> t;
        for(int i=0;i<S.size();i++){
```

```

        if (S[i]=='#') {
            if (s.size()>0) {
                s.pop_back();
            }
        } else {
            s.push_back(S[i]);
        }
    }
    for (int i=0;i<T.size();i++) {
        if (T[i]=='#') {
            if (t.size()>0) {
                t.pop_back();
            }
        } else {
            t.push_back(T[i]);
        }
    }
    if (s.size()!=t.size()) {
        return false;
    }
    for (int i=0;i<s.size();i++) {
        if (s[i]!=t[i]) {
            return false;
        }
    }
    return true;
}

};

```