

Edit Distance

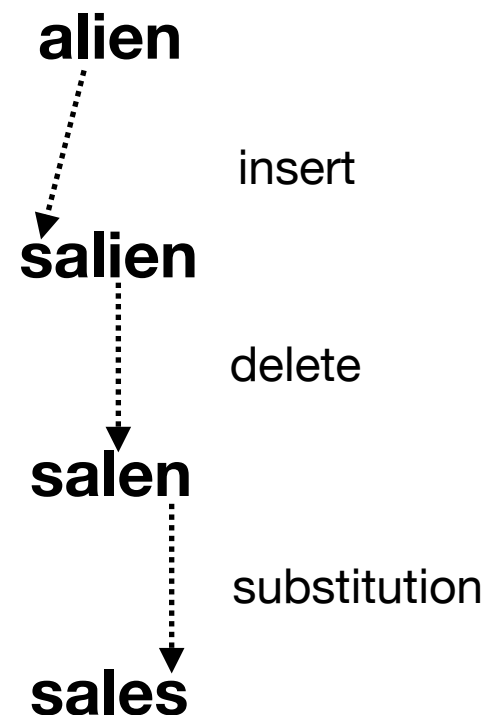
Use-it vs Lose-it

Edit Distance Problem

Given two strings word1 and word2, return the minimum number of operations required to convert word1 to word2. You have the following three operations permitted on a word:

- Replace a character
- Delete a character
- Insert a character

Edit Distance Example



```
>>> distance("alien", "sales")
```

```
3
```

```
>>> distance("spam", "poems")
```

```
4
```

Edit Distance

```
def distance(first, second):
```

```
    """ Returns the edit distance between first and second """
```

Edit Distance

```
def distance(first, second):  
    """ Returns the edit distance between first and second """  
    if first == '':  
        return len(second)
```

Edit Distance

```
def distance(first, second):  
    """ Returns the edit distance between first and second """  
    if first == '':  
        return len(second)  
    elif second == '':  
        return len(first)
```

Edit Distance

```
def distance(first, second):  
    """ Returns the edit distance between first and second """  
    if first == '':  
        return len(second)  
    elif second == '':  
        return len(first)  
    elif first[0] == second[0]:  
        return distance(first[1:], second[1:])
```

Edit Distance

```
def distance(first, second):  
    """ Returns the edit distance between first and second """  
    if first == '':  
        return len(second)  
    elif second == '':  
        return len(first)  
    elif first[0] == second[0]:  
        return distance(first[1:], second[1:])  
    else:
```


Edit Distance

```
def distance(first, second):  
    """ Returns the edit distance between first and second """  
    if first == '':  
        return len(second)  
    elif second == '':  
        return len(first)  
    elif first[0] == second[0]:  
        return distance(first[1:], second[1:])  
    else:  
        substitution = 1 + distance(first[1:], second[1:])
```

Edit Distance

```
def distance(first, second):  
    """ Returns the edit distance between first and second """  
    if first == '':  
        return len(second)  
    elif second == '':  
        return len(first)  
    elif first[0] == second[0]:  
        return distance(first[1:], second[1:])  
    else:  
        substitution = 1 + distance(first[1:], second[1:])  
        deletion = 1 + distance(first[1:], second)
```

Edit Distance

```
def distance(first, second):  
    """ Returns the edit distance between first and second """  
    if first == '':  
        return len(second)  
    elif second == '':  
        return len(first)  
    elif first[0] == second[0]:  
        return distance(first[1:], second[1:])  
    else:  
        substitution = 1 + distance(first[1:], second[1:])  
        deletion = 1 + distance(first[1:], second)  
        insertion = 1 + distance(first, second[1:])
```

Edit Distance

```
def distance(first, second):  
    """ Returns the edit distance between first and second """  
    if first == '':  
        return len(second)  
    elif second == '':  
        return len(first)  
    elif first[0] == second[0]:  
        return distance(first[1:], second[1:])  
    else:  
        substitution = 1 + distance(first[1:], second[1:])  
        deletion = 1 + distance(first[1:], second)  
        insertion = 1 + distance(first, second[1:])  
        return min(substitution, deletion, insertion)
```