

Distance d'édition

Algo naif

```
1 def SED(u, v):
2     if not len(u):
3         return len(v)
4     if not len(v):
5         return len(u)
6     pu = u[0:-1]
7     pv = v[0:-1]
8     if u[-1] == v[-1]:
9         return SED(pu, pv)
10    return 1 + min(SED(u, pv), SED(pu, v), SED(pu, pv))
```

Leveinshtein

Code from *Jill-Jenn Jie*.

```
1 n = len(x)
2 m = len(y)
3 A = [[i + j for j in range(m + 1)] for i in range(n + 1)]
4 for i in range(n):
5     for j in range(m):
6         A[i + 1][j + 1] = min(A[i][j + 1] + 1,           # insertion
7                               A[i + 1][j] + 1,           # suppress.
8                               A[i][j] + int(x[i] != y[j])) # substitut.
9 return A[n][m]
```