Distance d’edition

Algo naif

```python
def SED(u, v):
  if not len(u):
    return len(v)
  if not len(v):
    return len(u)
  pu = u[0:-1]
  pv = v[0:-1]
  if u[-1] == v[-1]:
    return SED(pu, pv)
  return 1 + min(SED(u, pv), SED(pu, v), SED(pu, pv))
```

Leveinshtein

Code from *Jill-Jenn Jie*.

```python
n = len(x)
  m = len(y)
  A = [[i + j for j in range(m + 1)] for i in range(n + 1)]
  for i in range(n):
    for j in range(m):
      A[i + 1][j + 1] = min(A[i][j + 1] + 1,              # insertion
                            A[i + 1][j] + 1,              # suppress.
                            A[i][j] + int(x[i] != y[j]))  # substitut.
  return A[n][m]
```