

21 NSI 05

```

In [ ]: def tri_insertion(L):
        n = len(L)

        # cas du tableau vide
        if ...:
            return L

        for j in range(1,n):
            e = L[j]
            i = j

            # A l'étape j, le sous-tableau L[0,j-1] est trié
            # et on insère L[j] dans ce sous-tableau en déterminant
            # le plus petit i tel que 0 <= i <= j et L[i-1] > L[j].
            while i > 0 and L[i-1] > ...:
                i = ...

            # si i != j, on décale le sous tableau L[i,j-1] d'un cran
            # vers la droite et on place L[j] en position i
            if i != j:
                for k in range(j,i,...):
                    L[k] = L[...]
```

📄

```

                L[i] = ...
        return L

```

