Sorting Algorithms

Julien BESTARD

Contents

1 Classique

Print Algorithm

```
def printmat(M):
    dim1 = len(M[0])
    for i in range(len(M)):
        for j in range(dim1):
            print(M[i][j], end ="␣")
        print()
```

PrettyPrint Algorithm

```
def prettyprint(M,d):
    s = "|{:"+str(d)+"d}"
    dim1 = len(M[0])
    for i in range(len(M)):
        print("-"*(d+2)*dim1)
        for j in range(dim1):
            print(s.format(M[i][j]), end="␣")
        print("|")
    print("-"*(d+2)*dim1)
```

Init & Load Algorithm

```
def init(l,c,val):
    matrix = []
    for i in range(l):
        l = []
        for j in range(c):
            l.append(val)
        matrix.append(l)
    return matrix

def __str2intlist(str):
    L = []
    for c in str:
        nb = ""
        if c != "␣":
            nb += c
        else:
            L.append(nb)
    return L

def load(filename):
    f = open(filename)
    lines = f.readlines()
    f.close()
    M = []
    for line in lines:
        M.append(__str2intlist(line))
    return M
```

Add Matrix Algorithm

```
def add_matrices(A,B):
    dim0 = len(A)
    dim1 = len(A[0])
    if (dim0 == len(B)) and (dim1 == len(B[0])):
        M = []
        for i in range(dim0):
            L = []
            for j in range(dim1):
                L.append(A[i][j] + B[i][j])
            M.append(L)
    else:
        raise Excepetion "Matrix␣need␣to␣be␣of␣same␣length"
```