# Sorting Algorithms

Julien BESTARD

# Contents

# 1 Classique

Print Algorithm

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

PrettyPrint Algorithm

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

### 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 = []
    nb = ""
    for c in str:
        if c != " " or c != "\n":
            nb += c
        else:
            L.append(nb)
    return L

def __str2intlist2(str):
    n = len(s)-1
    i = 0
    L = []
    while i<n:
        word = ""
        while i<n and L[i]!=" ":
            word += str[i]
            i += 1
        L.append(int(word))
        i += 1
    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):
    l = len(A)
    c = len(A[0])
    if (l == len(B)) and (c == len(B[0])):
        M = init(l,c,0)
        for i in range(l):
            for j in range(c):
                M[i][j] = A[i][j] + B[i][j]
        return M
    else:
        raise Exception("Matrix Error")
```

### Mult Matrix Algorithm

```
def mult_matrices(A,B):
    m = len(A)
    n = len(A[0])
    p = len(B[0])
    if (n != len(B)):
        raise Exception("Matrix Error")
    M = init(m,p,0)
    for i in range(m):
        for j in range(p):
            for k in range(n):
                M[i][j] += A[i][k] * B[k][j]
    return M
```

## 2 Recherche et test

MaxGap

```
def maxGap(M)
    l = len(M)
    c = len(M[0])
    if (l >= 2) and (c >= 2):
        gap = []
        for i in range(l):
            min = M[i][0]
            max = M[i][0]
            for j in range(1,c):
                if M[i][j] < min:
                    min = M[i][j]
                elif M[i][j] >= max:
                    max = M[i][j]
            gap.append(max - min)
        max = 0
        for elt in gap:
            if elt >= max:
                max = elt
        return max

def maxGap(M):
    l = len(M)
    c = len(M[0])
    if (l >= 2) and (c >= 2):
        gap = []
        for i in range(l):
            maxG = max(M[i])
            minG = min(M[i])
            gap.append(maxG - minG)
        return max(gap)
    else:
        raise Exception("Matrix not conform")
```