****

Algorithmique

Informatique, Classe I1A

**S07 : Quicksort, Mergesort, Divide-and-conquer**

Date : 29/03/2018

Noms : Luke Perrottet, Stefan Pahud

Table des matières

[1 Ex. 1 2](#_Toc510098506)

[1.1 Code 2](#_Toc510098507)

[1.2 Résultat 2](#_Toc510098508)

[2 Ex. 2 3](#_Toc510098509)

[2.1 Code 3](#_Toc510098510)

[2.2 Résultat 3](#_Toc510098511)

[3 Ex. 3 3](#_Toc510098512)

[3.1 Code 3](#_Toc510098513)

[3.2 Résultat 3](#_Toc510098514)

[3.2.1 myFile.txt 3](#_Toc510098515)

# Ex. 1

## Code

**package** s07;

**public** **class** **Ex01** {

**public** **static** **int** **minUniqueCall**(**int**[] table) {

**return** **minUniqueCall**(table, table.length - **1**);

}

**public** **static** **int** **minUniqueCall**(**int**[] table, **int** index) {

**if** (index < **0**)

**return** Integer.MAX\_VALUE;

**return** Math.min(table[index], minUniqueCall(table, index - **1**));

}

**public** **static** **int** **minDoubleCall**(**int**[] table) {

**return** **minDoubleCall**(table, **0**, table.length - **1**);

}

**public** **static** **int** **minDoubleCall**(**int**[] table, **int** left, **int** right) {

**int** mid;

**if** (left >= right)

**return** table[left];

mid = (left + right) / **2**;

**return** Math.min(minDoubleCall(table, left, mid), minDoubleCall(table, mid + **1**, right));

}

**public** **static** **void** **main**(String[] args) {

**int**[] tab = { -**1**, **5**, **8**, -**12**, **1**, **9**, **6**, -**1**, **10** };

System.out.println("Unique call : " + minUniqueCall(tab));

System.out.println("Double call : " + minDoubleCall(tab));

}

}

## Résultat

Unique call : -**12**

Double call : -**12**

# Ex. 2

## Code

**private** **static** **int** **partition**(**int**[] t, **int** left, **int** right) {

**int** p = t[left];

**int** k = left;

**for** (**int** i = left + **1**; i <= right; i++) {

**if** (t[i] <= p) {

**int** lower = t[i];

t[i] = t[k + **1**];

t[++k] = lower;

}

}

t[left] = t[k];

t[k] = p;

**return** k;

}

## Résultat

OK. Tiny test passed...

# Ex. 3

## Code

**private** **static** **int** **split**(String a, String b, String c) **throws** IOException {

BufferedReader fa = **new** BufferedReader(**new** FileReader(a));

PrintWriter fb = **new** PrintWriter(**new** FileWriter(b));

PrintWriter fc = **new** PrintWriter(**new** FileWriter(c));

String line = fa.readLine();

String prevLine = **null**;

**boolean** writeInB = **true**;

**int** m = **1**;

**while** (line != **null**) {

**if** (!isMonotone(line, prevLine)){

writeInB = !writeInB;

m++;

}

**if** (writeInB)

fb.println(line);

**else**

fc.println(line);

prevLine = line;

line = fa.readLine();

}

fa.close();

fb.close();

fc.close();

**return** m;

}

## Résultat

### myFile.txt

ananas

bal

canard

maison

zebra