L3/M1, 2005–2006

# Algorithmique effective – DM 1

Sylvain.Perifel@ens-lyon.fr

## 1 Pour les L3

À faire seul. Rendre un petit rapport (1 page, tapé avec LaTeX) expliquant la structure de données, le principe de l'algorithme, ainsi que le pseudo-code de l'algorithme. Vous enverrez par courriel le code source (`votre_nom.cpp`) et le rapport au format Postscript (`votre_nom.ps`).

**Exercice 1**
PC : 110401 ; UVa : 10041.
The famous gangster Vito Deadstone is moving to New York. He has a very big family there, all of them living on Lamafia Avenue. Since he will visit all his relatives very often, he wants to find a house close to them.

Indeed, Vito wants to minimize the total distance to all of his relatives and has blackmailed you to write a program that solves his problem.

*Input*
The input consists of several test cases. The first line contains the number of test cases.

For each test case you will be given the integer number of relatives $r$ ($0 < r < 500$) and the street numbers (also integers) $s_1, s_2, \ldots, s_i, \ldots, s_r$ where they live ($0 < s_i < 300000$). Note that several relatives might live at the same street number.

*Output*
For each test case, your program must write the minimal sum of distances from the optimal Vito's house to each one of his relatives. The distance between two street numbers $s_i$ and $s_j$ is $d_{i,j} = |s_i - s_j|$.

*Sample Input*

```
2
2 2 4
3 2 4 6
```

*Sample Output*

```
2
4
```

## 2 Pour les M1

À faire par groupe de trois. Rendre un petit rapport (1 page, tapé avec LaTeX) expliquant la structure de données, le principe de l'algorithme, ainsi que le pseudo-code de l'algorithme. Vous enverrez par courriel le code source (`nom1_nom2_nom3.cpp`) et le rapport en Postscript (`nom1_nom2_nom3.ps`).

Pour les bonnes volontés parmi les L3, une référence sur les arbres couvrants minimaux est *Introduction to algorithms*, de Cormen, Leiserson, Rivest, Shamir, à la page 561 (chapitre 23) de la seconde édition.

**Exercice 2**
PC : 111001 ; UVa : 10034.
In an episode of the Dick Van Dyke show, little Richie connects the freckles on his Dad's back to form a picture of the Liberty Bell. Alas, one of the freckles turns out to be a scar, so his Ripley's engagement falls through.

Consider Dick's back to be a plane with freckles at various $(x, y)$ locations. Your job is to tell Richie how to connect the dots so as to minimize the amount of ink used. Richie connects the dots by drawing straight lines between pairs, possibly lifting the pen between lines. When Richie is done there must be a sequence of connected lines from any freckle to any other freckle.

*Input*
The input begins with a single positive integer on a line by itself indicating the number of test cases, followed by a blank line.

The first line of each test case contains $0 < n < 100$, giving the number of freckles on Dick's back. For each freckle, a line follows ; each following line contains two real numbers indicating the $(x, y)$ coordinates of the freckle.

There is a blank line between each two consecutive test cases.

*Output*
For each test case, your program must print a single real number to two decimal places : the minimum total length of ink lines that can connect all the freckles. The output of each two consecutive cases must be separated by a blank line.

*Sample Input*

```
1

3
1.0 1.0
2.0 2.0
2.0 4.0
```

*Sample Output*

```
3.41
```

## 3 Optionnel

**Exercice 3**
PC : 110407 ; UVa : 10152.
King Yertle wishes to rearrange his turtle throne to place his highest-ranking nobles and closest advisors nearer to the top. A single operation is available to change the order of the turtles in the stack : a turtle can crawl out of its position in the stack and climb up over the other turtles to sit on the top.

Given an original ordering of a turtle stack and a required ordering for the same turtle stack, your job is to find a minimal sequence of operations that rearranges the original stack into the required stack.

*Input*

The first line of the input consists of a single integer $K$ giving the number of test cases. Each test case consists of an integer $n$ giving the number of turtles in the stack. The next $n$ lines describe the original ordering of the turtle stack. Each of the lines contains the name of a turtle, starting with the turtle on the top of the stack and working down to the turtle at the bottom of the stack. Turtles have unique names, each of which is a string of no more than eighty characters drawn from a character set consisting of the alphanumeric characters, the space character and the period ("."). The next $n$ lines in the input give the desired ordering of the stack, once again by naming turtles from top to bottom. Each test case consists of exactly $2n + 1$ lines in total. The number of turtles ($n$) will be less than or equal to 200.

*Output*

For each test case, the output consists of a sequence of turtle names, one per line, indicating the order in which turtles are to leave their positions in the stack and crawl to the top. This sequence of operations should transform the original stack into the required stack and should be as short as possible. If more than one solution of shortest length is possible, any of the solutions may be reported.

Print a blank line after each test case.

*Sample Input*

```
2
3
Yertle
Duke of Earl
Sir Lancelot
Duke of Earl
Yertle
Sir Lancelot
9
Yertle
Duke of Earl
Sir Lancelot
Elizabeth Windsor
Michael Eisner
Richard M. Nixon
Mr. Rogers
Ford Perfect
Mack
Yertle
Richard M. Nixon
Sir Lancelot
Duke of Earl
Elizabeth Windsor
Michael Eisner
Mr. Rogers
Ford Perfect
Mack
```

*Sample Output*

Duke of Earl

Sir Lancelot
Richard M. Nixon
Yertle

⋆

⋆ ⋆