Algorithmique I
Année 2007-2008

A. Benoit, B. Depardon
JF. Pineau, C. Rezvoy

Analysis of Algorithms

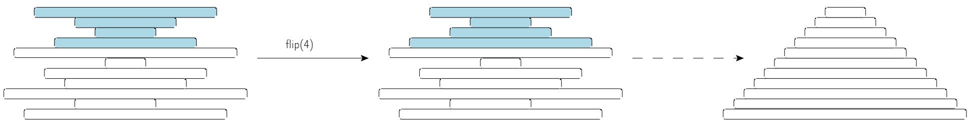
Midterm Exam

October 24^{th}, 2007

2 Heures

1 Pancakes

You are given a pile of pancakes that need to be sorted by size. The only operation that can be done consists in using a spatula to flip the pile. We define the function $flip(k)$ that consists in inserting the spatula under the top k pancakes and flip the top of the pile over, as illustrated below.



Question 1.1

Devise an algorithm that sorts a pile of pancake solely using the operation $flip$. Define the worst case complexity of your algorithm in number of calls to $flip$.

Question 1.2 Justify the correctness of your algorithm.

Question 1.3 The pancakes that need sorting now each have one of their side burnt. To tidy things up a little your are asked to sort the pancakes while making sure that the burnt side of each pancake is facing down. What would then be the complexity of your algorithm ?

2 Brass Band

A brass band is composed of n musicians of sizes $s_1, s_2, \ldots, s_n$. For parades, the band has a set of m uniforms ($m \geq n$) of sizes $u_1, u_2, \ldots, u_m$. Each year some musicians leave the band and are replaced by new ones. Uniforms must then be redistributed amongst musicians so that each player finds the uniform that suits him best.

Clément, the drummer, thinks the most efficient way to do so is to minimize the mean difference between the size of a player and the size of its uniform :

$$\frac{1}{n}\sum_{i=1}^{n}|t_i - u_{\alpha(i)}|$$

where $\alpha(i)$ is the index of the uniform attributed to the player of size t_i. To do so, he devises a greedy algorithm that consists in looking for the pair i, j minimizing $|s_i - u_j|$. The musician i is attributed the uniform j and the process iterates until everyone has received an uniform.

Question 2.1 Is Clément's algorithm optimal ?

Anne, the horn player, thinks it would be fairer to try to minimize the mean square difference between the size of a player and that of its uniform :

$$\frac{1}{n}\sum_{i=1}^{n}(t_i - u_{\alpha(i)})^2$$

Question 2.2 Show by an example the advantage this objective function has over the previous one. Is the greedy algorithm optimal for this new objective function ?

Question 2.3 Show that if there are as many uniforms as musicians, the algorithm consisting in sorting musicians and uniforms by size and assigning the uniform i to the musician i is optimal for the second objective function.

Question 2.4 Give an algorithm returning an optimal solution for any m, $m \geq n$.

3 Sub-vector of maximum sum

Given a table of size T of n relative integers, the goal is to find $max\{\ \forall i, j \in \{1 \dots n\} \mid \sum_{k=i}^{j} T[k]\ \}$. For instance, given the following table :

2	18	−22	**20**	**8**	**−6**	**10**	−24	13	3

the algorithm would return the sum of elements 4 to 7 *i.e.* 32.

Question 3.1 Devise an algorithm returning the maximum sum of contiguous elements following a *divide and conquer* scheme.

Question 3.2 Devise an algorithm returning the maximum sum of contiguous elements following a *dynamic programming* approach.

Question 3.3 Compare the worst case complexity of the two algorithms in terms of number of comparisons.

4 The library

The library is moving into a new building. Its collection consists in n books $b_1, b_2, \dots, b_n$. The book b_i has a width w_i and a height h_i. The books should be stored in a given order (by increasing i) on identical bookshelves of width L.

Question 4.1 Supposing each book has the same height $h = h_i, 1 \leq i \leq n$, show that the greedy algorithm which stores the books side by side until no more can fit minimizes the number of required bookshelves.

Question 4.2 Now let us assume that books have different heights but the height between the shelves can be adjusted. The goal is to minimize the overall height defined as the sum of the heights of the tallest book on each bookshelf.

4.2.1 Give an example showing that the previously defined greedy algorithm does not always yield the minimal overall height.

4.2.2 Devise an algorithm that optimally solves this problem.

Question 4.3 We go back to the case where all the books have the same height $h = h_i, 1 \leq i \leq n$. The goal now is to store the n books on k bookshelves of length L (k being a parameter of the problem) while minimizing L. Devise an algorithm to solve that problem, and express its complexity as a function of n and L.

5 Bonus (difficult)

Question 5.1 Given a set S of $n \geq 2$ distinct unsorted integers, devise an $O(n)$ algorithm that finds two distinct elements x and y of S so that $|x - y| \leq \frac{1}{n-1}(\max(S) - \min(S))$.