Propabilistic Aspects of Computer Science, DM1

Exercice 1 **Analysis of the Quick-sort algortihm**

Algorithme 1 : Quick_sort

Input : A list S of n dictinct numbers
Output : The sorted list of the elements of S
begin
  if *S has 0 or 1 element* **then** return S;
  else
    Choose an element x (pivot) of S and separate the other elements in tao sub-lists
    — S_1, list of the elements of S that are $< x$;
    — S_2, list of the elements of S that are $> x$;
    Quick_sort($S1$) ; Quick_sort(S_2);
    Return the list S_1, x, S_2.

1. Give an example of a family of lists that requires $\Omega(n^2)$ comparisons to sort the lists with this algorithm.

The goal of the exercise is to show that if the pivots are chosen uniformly at random, then the expectation of the number of comparicons is $2n \ln n + O(n)$. We note $y_1 < y_2 < \cdots < y_n$ the elements of the list.

2. What is the probability that two elements y_i and y_j are compared during the first stage of the algorithm, i.e. before the second pivot is chosen, under the condition that the first pivot is in $[y_i, y_j]$?

3. What is the probability that two elements y_i and y_j are compared at some point of the whole computation ? One could prove the result by recurrence on something.

4. Deduce the result.

5. What happens if the first element is always chosen as pivot ? What is the difference with the choice of a random pivot ?