*Algorithms and programming*
École normale supérieure - Département d'informatique
TD 4: Sorting algorithms

---

1. Let $b_n = \frac{1}{n+1}\binom{2n}{n}$ ($b_n$ is the $n^{th}$ Cathalan number). Show that $b_{n+1} = \sum_{i=0}^{i=n} b_i b_{n-i}$.
2. Give an $O(n \log k)$ algorithm to merge $k$ sorted lists into one sorted list, where $n$ is the total length of the lists.
3. Show that the second smallest of $n$ elements can be found with $n + \lceil \log n \rceil - 2$ comparisons in the worst case.
4. Show that there is no comparison sort whose running time is linear for at least half of the $n!$ inputs of length $n$. What about a fraction of $1/n$ of the inputs of length $n$ ? What about a fraction $1/2^n$ ?
5. Show a lower bound on the running time of an algorithm merging two sorted lists containing $n$ elements each.
   (a) Show that there are $\binom{2n}{n}$ ways to divide $2n$ numbers into two sorted lists of length $n$.
   (b) Using a decision tree show that any algorithm merging two sorted lists makes $2n - o(n)$ comparisons.
   (c) Show that if two elements consecutive in the merged list come from two different input lists, then the algorithm had to compare them.
   (d) Show that any merge algorithm has to make at least $2n - 1$ comparisons in the worst case.
6. Show that $\lceil 3n/2 \rceil - 2$ comparisons are necessary in the worst case to find both the maximum and minimum of n numbers.
7. Describe an algorithm that, given $n$ integers in the range 1 to $k$, preprocesses its input and then answers any query about how many of the $n$ integers fall into a range $[a, b]$ in $0(1)$ time.
8. Let $A[1..n]$ be an array of $n$ distinct integers. If $i < j$ and $A[i] > A[j]$ we call $(i, j)$ an *inversion*.
   (a) Find all inversions in the array $< 2, 3, 8, 6, 1 >$
   (b) Build an array out of a set $\{1, 2, \dots, n\}$ containing the maximum possible set of inversions. How many inversions does this array contain ?
   (c) What is the relation between the running time of insertion sort and the number of inversions in an array ?
   (d) Develop an algorithm to count the number of inversions in an array. The algorithm must use $O(n \log n)$ time.
9. Suppose that we have an array of $n$ data records to sort and that the key of each record has the value 0 or 1.

(a) Give a simple, linear-time algorithm for sorting the $n$ data records in place. Use no storage of more than constant size in addition to the storage provided by the array.

(b) Can your sort from part (a) be used to radix sort $n$ records with $b$-bit keys in $O(bn)$ time ? Explain how or why not.

10. Let $A$ be an array of strings. The length of strings in $A$ is not fixed, but their total length is $n$. Sort $A$ in $O(n)$ time. The order on strings in alphabetic, for example, $a < ab < b$.
11. In an ordered array of odd length, the median is the middle value. If the length of the array is even, the median is the mean of the two middle values. Design a data structure that supports the following two operations : $add(x)$ (add $x$ to the data structure) and median (returns the median of all elements so far).
12. Given an array of integers, there is a sliding window of size k which is moving from the very left of the array to the very right, one position at a time. You can only see the $k$ integers in the window. Develop an algorithm that outputs the minimum integer in the window for each position. Can you develop an $O(n)$-time algorithm ?