



LSINF1121 ALGORITHMIQUE ET
STRUCTURES DE DONNÉES

TP2: TRIS ET PROPRIÉTÉS DES ENSEMBLES TRIÉS

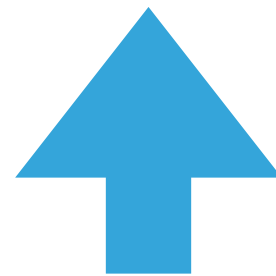
QUESTION 2.1.1: COMMENT TROUVER OÙ INSÉRER DANS UN TABLEAU TRIÉ?

1		3		6		10		15		21		27		34		42		51
---	--	---	--	---	--	----	--	----	--	----	--	----	--	----	--	----	--	----

Où insérer 30?

QUESTION 2.1.1: COMMENT TROUVER OÙ INSÉRER DANS UN TABLEAU TRIÉ?

1		3		6		10		15		21		27		34		42		51		61
---	--	---	--	---	--	----	--	----	--	----	--	----	--	----	--	----	--	----	--	----

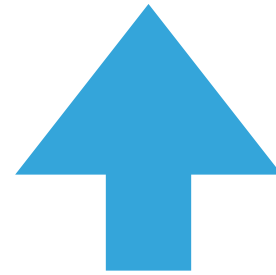


$21 < 30$: c'est à droite!

Où insérer 30?

QUESTION 2.1.1: COMMENT TROUVER OÙ INSÉRER DANS UN TABLEAU TRIÉ?

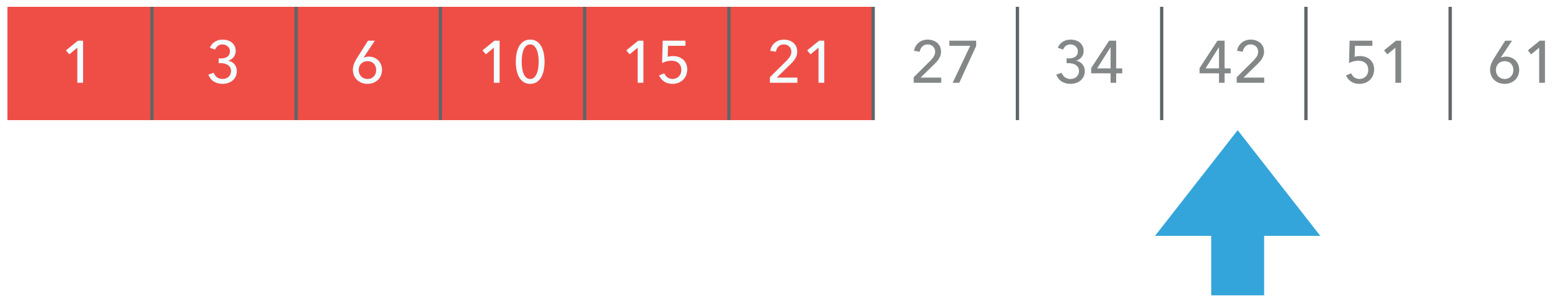
1	3	6	10	15	21	27	34	42	51	61
---	---	---	----	----	----	----	----	----	----	----



$21 < 30$: c'est à droite!

Où insérer 30?

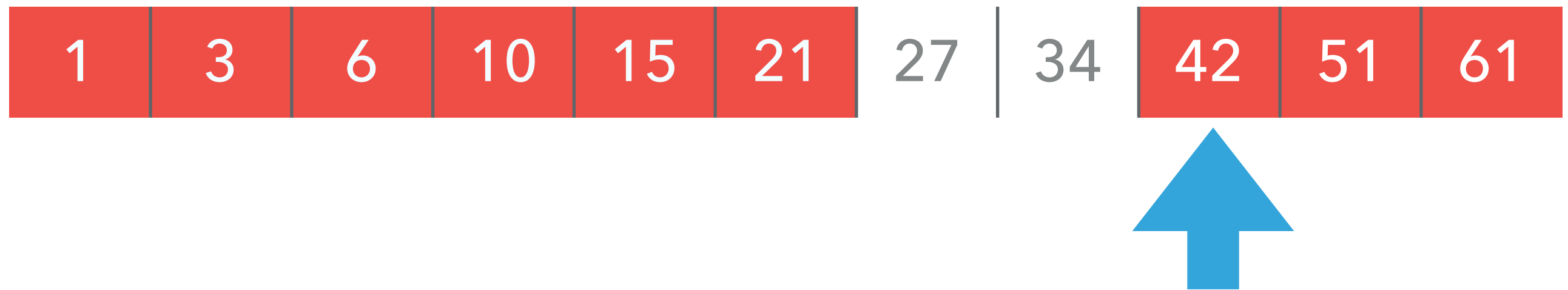
QUESTION 2.1.1: COMMENT TROUVER OÙ INSÉRER DANS UN TABLEAU TRIÉ?



$42 > 30$: c'est à gauche!

Où insérer 30?

QUESTION 2.1.1: COMMENT TROUVER OÙ INSÉRER DANS UN TABLEAU TRIÉ?

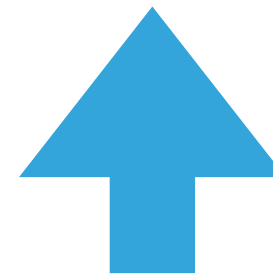


$42 > 30$: c'est à gauche!

Où insérer 30?

QUESTION 2.1.1: COMMENT TROUVER OÙ INSÉRER DANS UN TABLEAU TRIÉ?

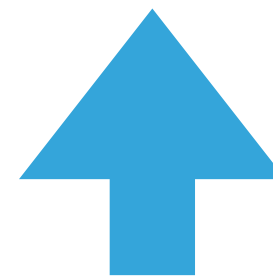
1	3	6	10	15	21	27	34	42	51	61
---	---	---	----	----	----	----	----	----	----	----



$27 < 30$: c'est à droite!

Où insérer 30?

QUESTION 2.1.1: COMMENT TROUVER OÙ INSÉRER DANS UN TABLEAU TRIÉ?

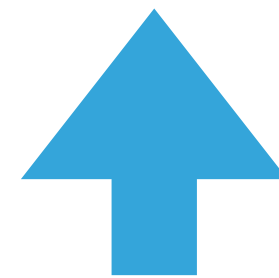


$27 < 30$: c'est à droite!

Où insérer 30?

QUESTION 2.1.1: COMMENT TROUVER OÙ INSÉRER DANS UN TABLEAU TRIÉ?

1	3	6	10	15	21	27	34	42	51	61
---	---	---	----	----	----	----	----	----	----	----



$34 < 30$: c'est à gauche!

Où insérer 30?

QUESTION 2.1.1: COMMENT TROUVER OÙ INSÉRER DANS UN TABLEAU TRIÉ?

Quelle est la complexité?

$$f(n) = 1 + f\left(\frac{n}{2}\right)$$
$$f(1) = 0$$

$$f(n) = \log_2 n$$

QUESTION 2.1.1: COMMENT TROUVER OÙ INSÉRER DANS UN TABLEAU TRIÉ?

Est-il possible d'aller encore plus vite?

Oui, si on a des informations sur la répartition des éléments.

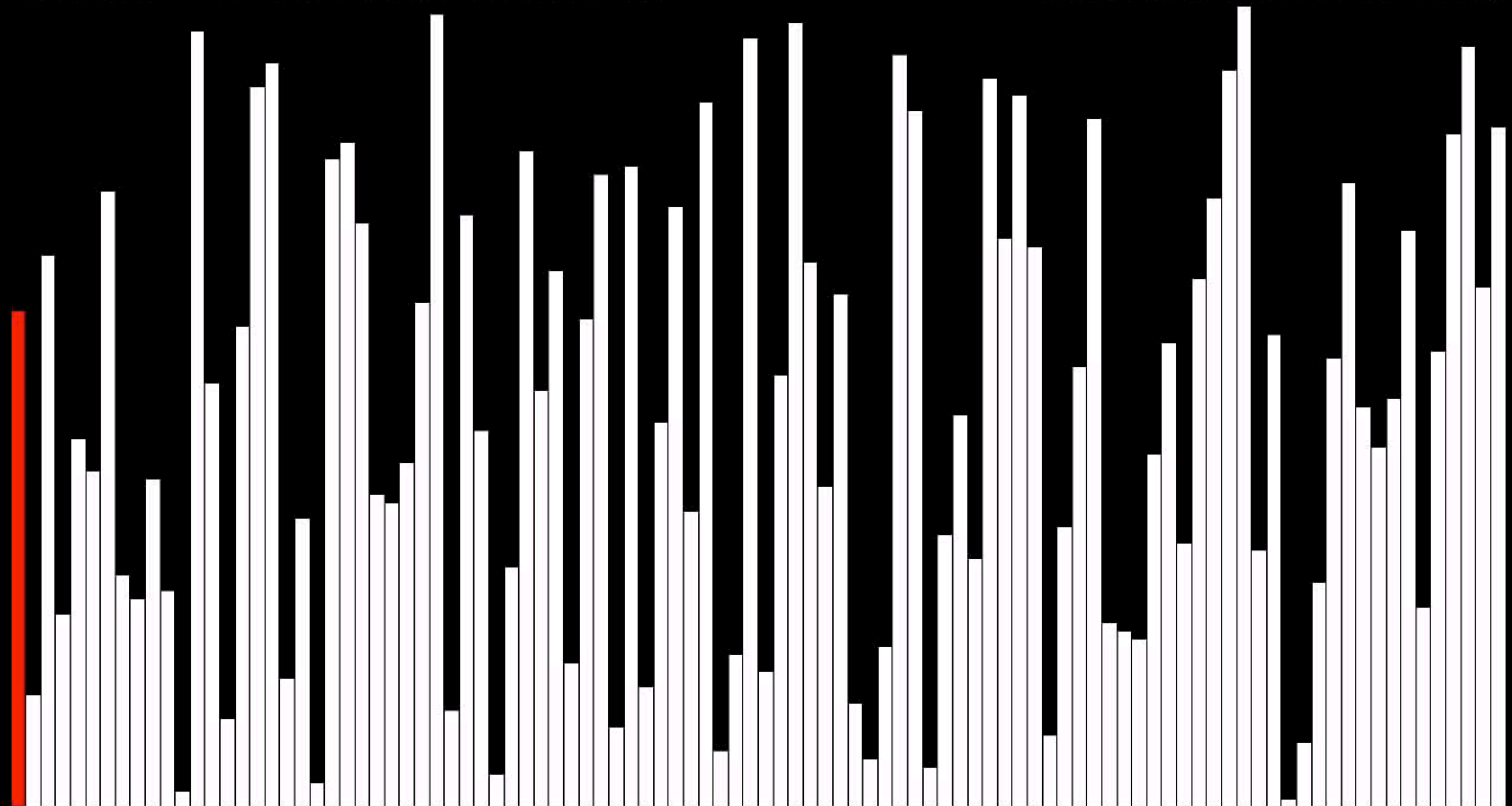
Flamenco binary search



LSINF1121 TP2: TRIS ET PROPRIÉTÉS DES ENSEMBLES TRIÉS

Selection Sort - 0 comparisons, 1 array accesses, 60 ms delay

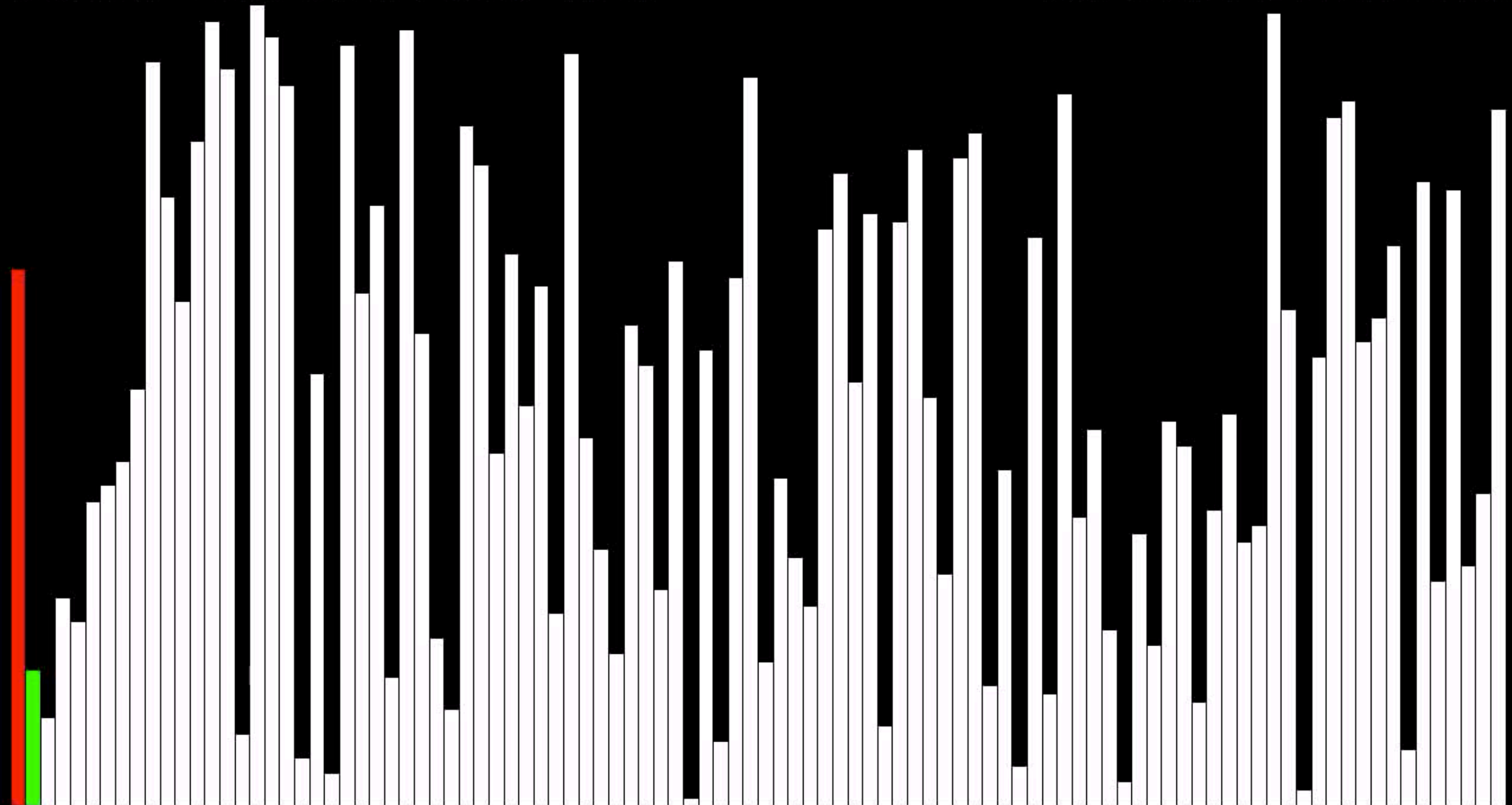
<http://panthema.net/2013/sound-of-sorting>



LSINF1121 TP2: TRIS ET PROPRIÉTÉS DES ENSEMBLES TRIÉS

Insertion Sort - 0 comparisons, 2 array accesses, 31 ms delay

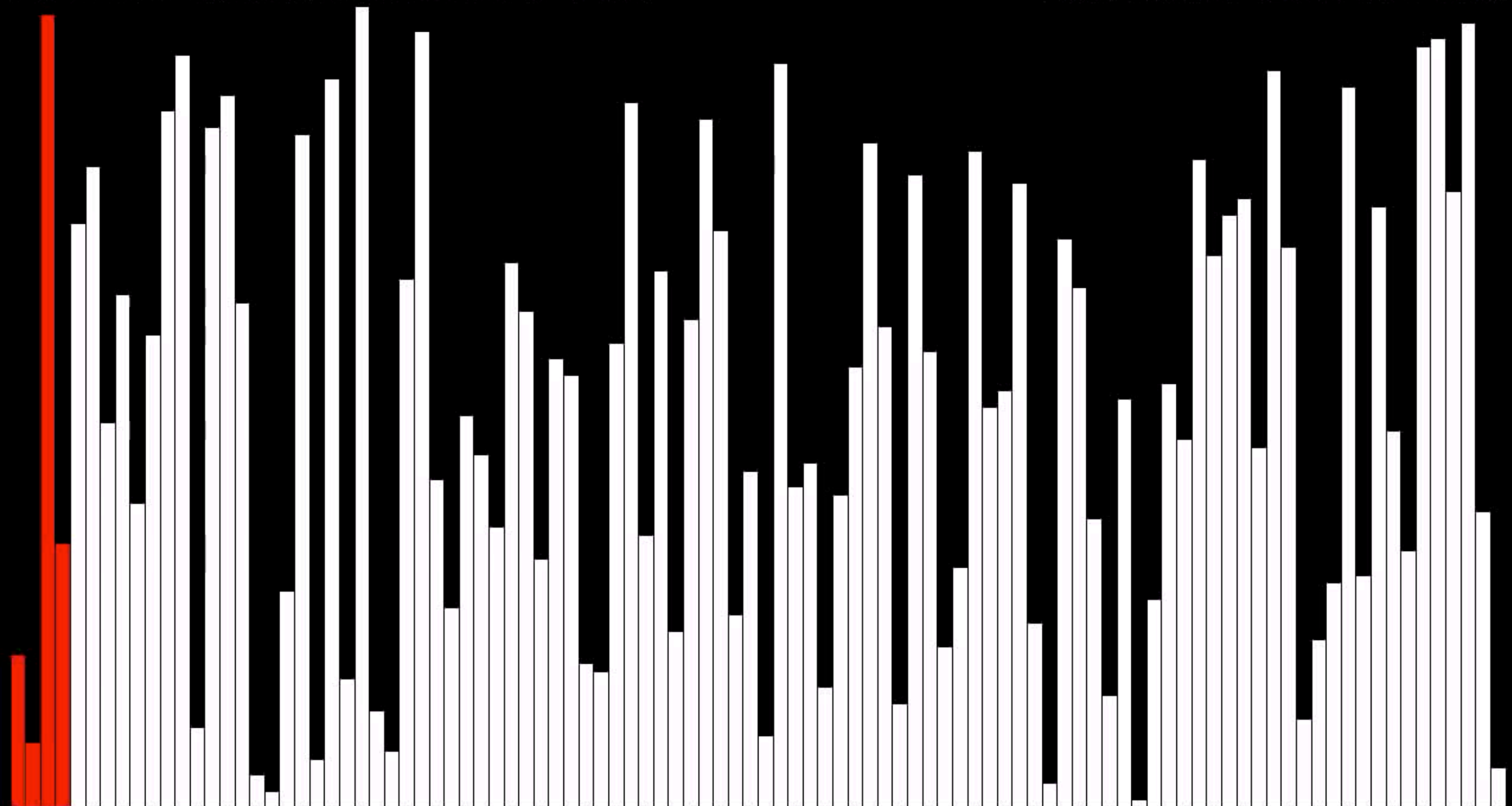
<http://panthema.net/2013/sound-of-sorting>



LSINF1121 TP2: TRIS ET PROPRIÉTÉS DES ENSEMBLES TRIÉS

Bubble Sort - 3 comparisons, 9 array accesses, 4.0 ms delay

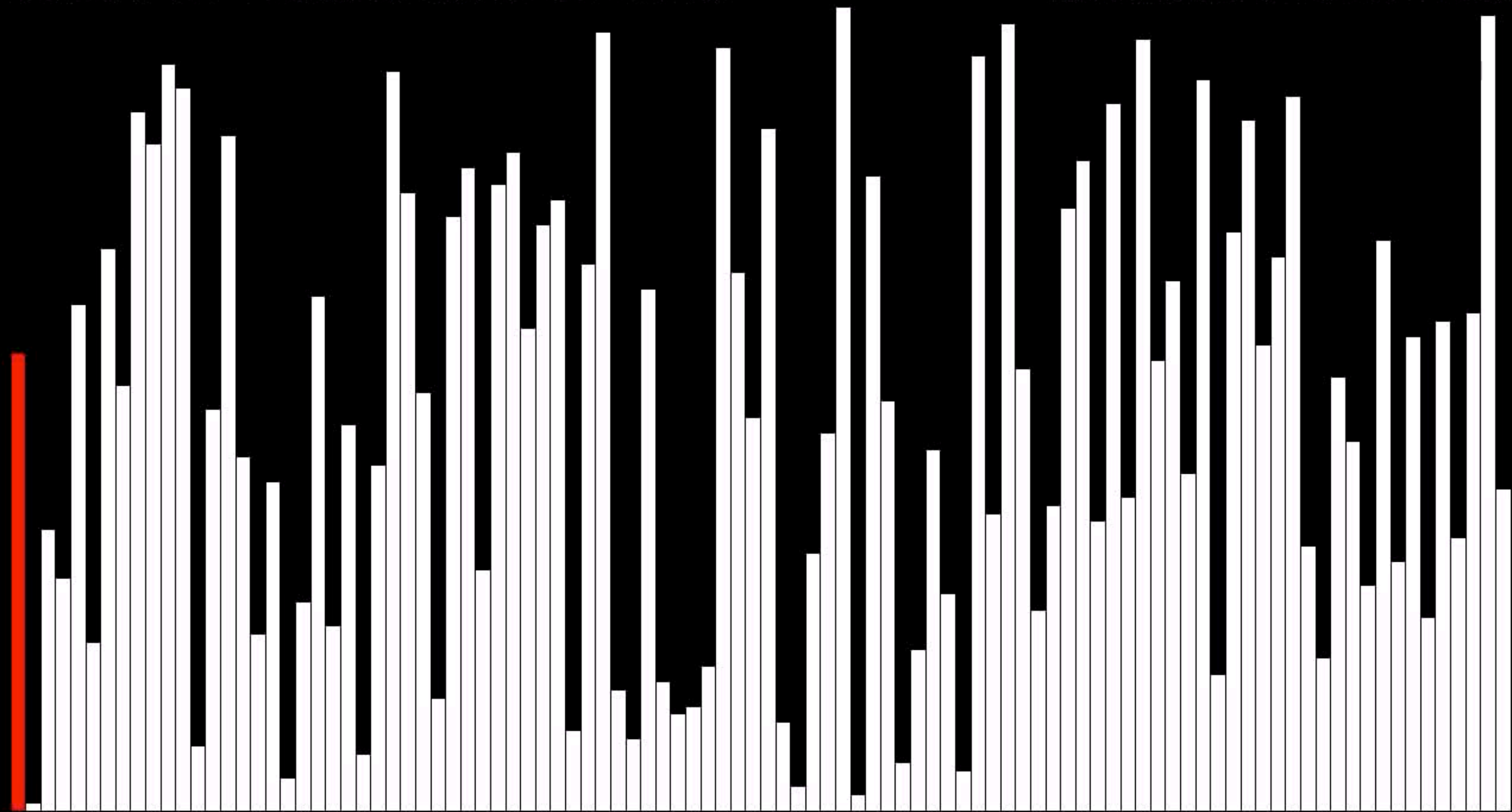
<http://panthema.net/2013/sound-of-sorting>



LSINF1121 TP2: TRIS ET PROPRIÉTÉS DES ENSEMBLES TRIÉS

Quick Sort (LL ptrs) - 0 comparisons, 1 array accesses, 50 ms delay

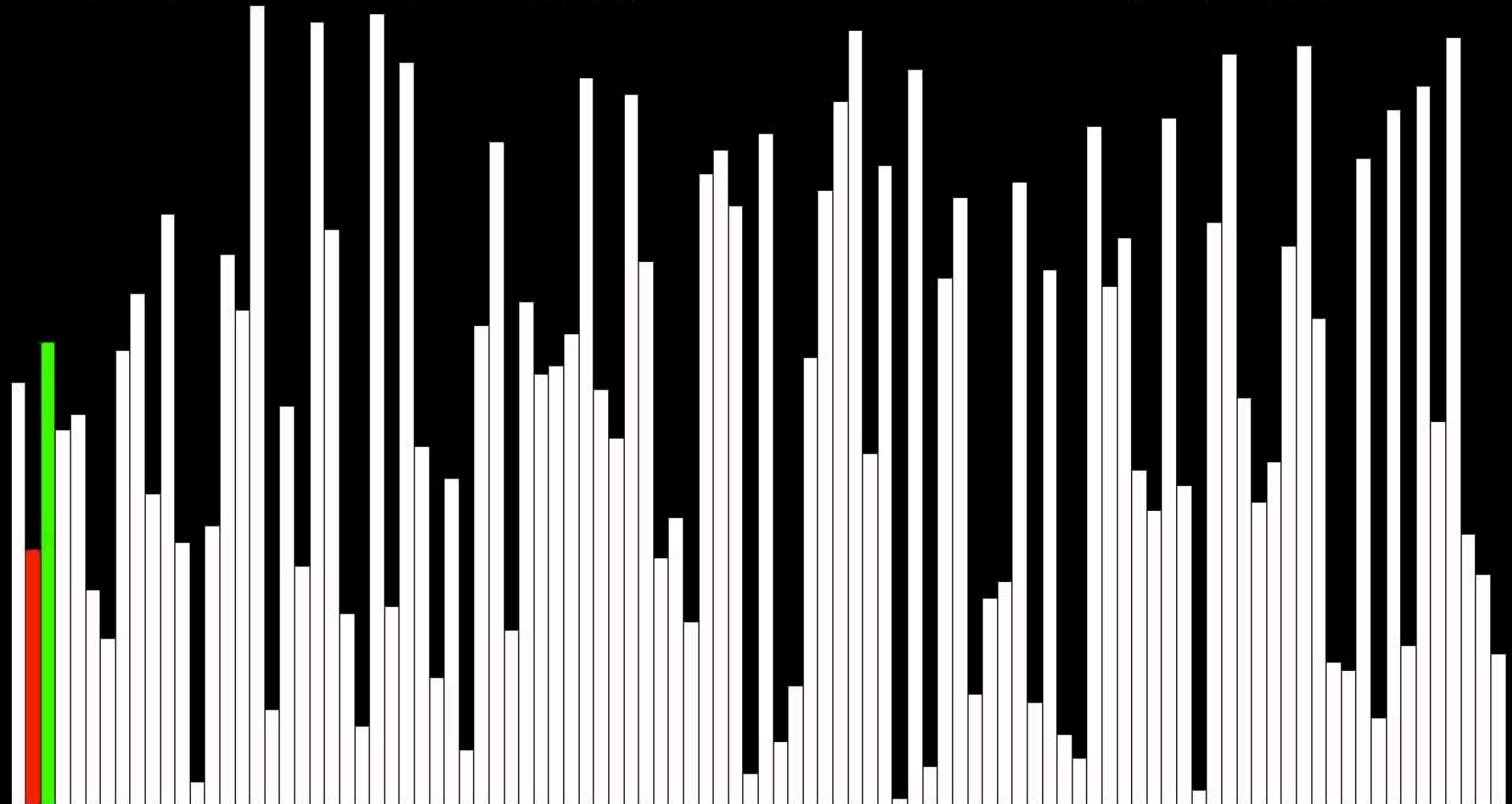
<http://panthema.net/2013/sound-of-sorting>



LSINF1121 TP2: TRIS ET PROPRIÉTÉS DES ENSEMBLES TRIÉS

Merge Sort - 0 comparisons, 1 array accesses, 35 ms delay

<http://panthema.net/2013/sound-of-sorting>



Shell sort

Forces et faiblesses - Random

Selection Sort

Shell Sort

Insertion Sort

Merge Sort

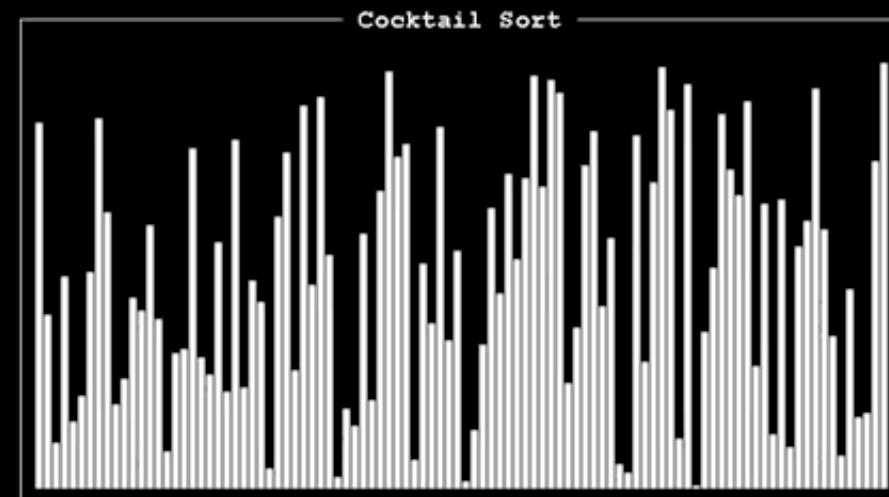
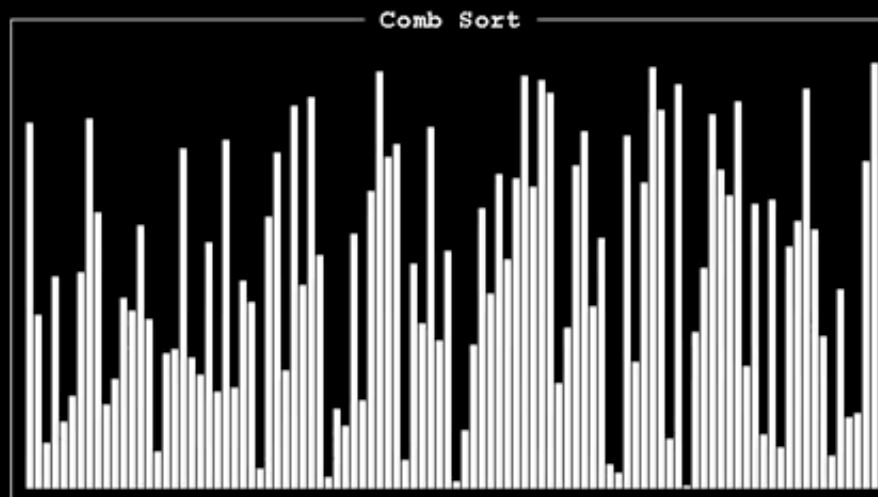
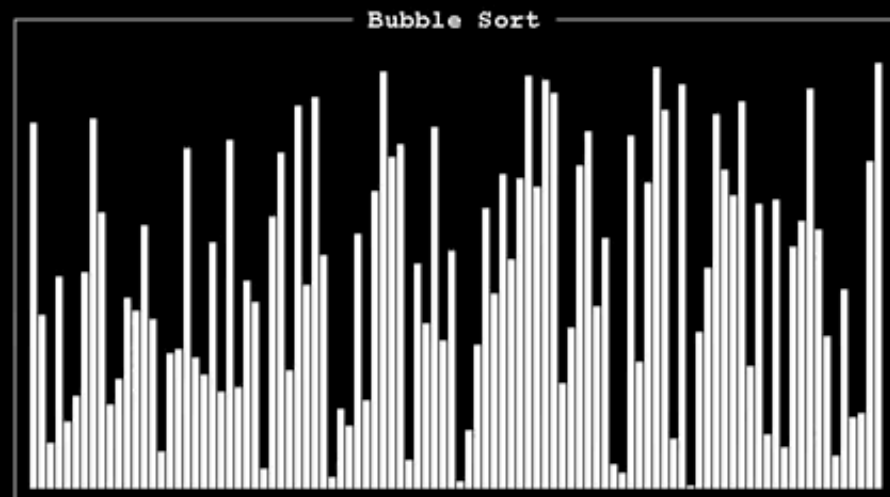
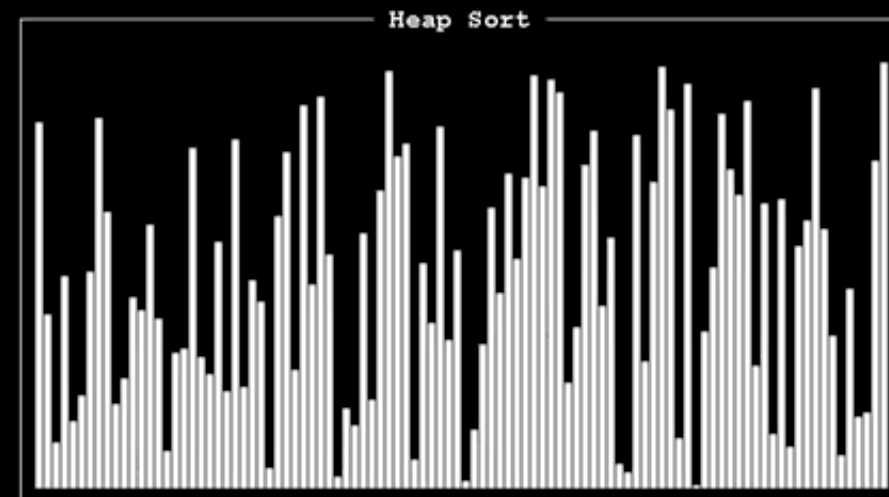
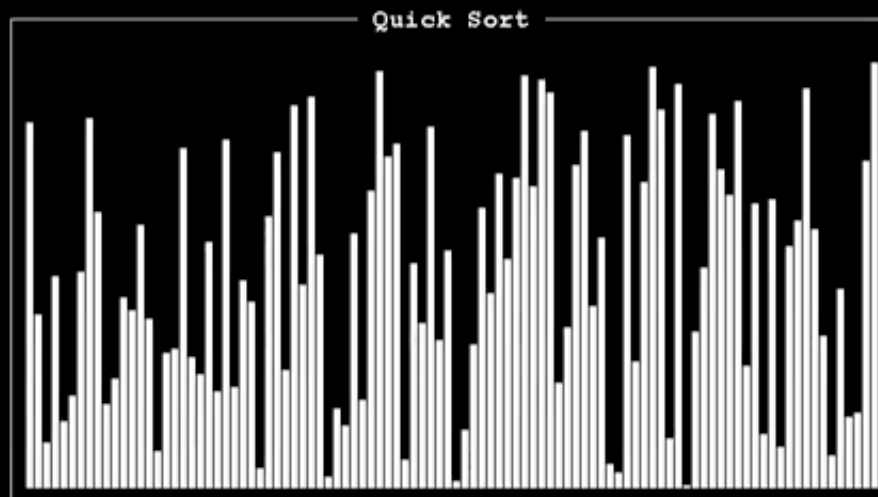
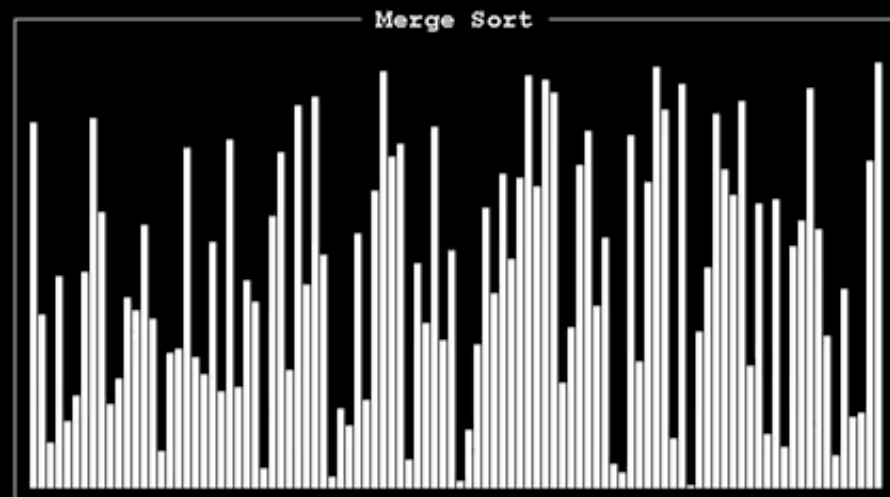
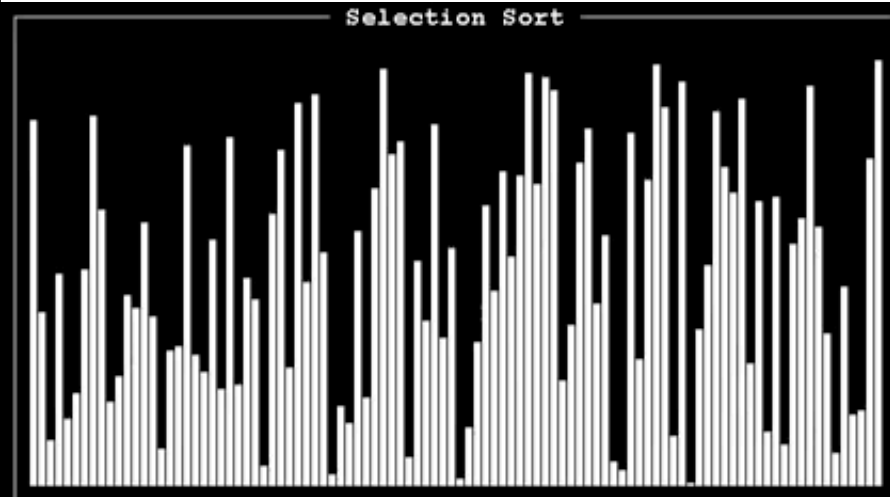
Quick Sort

Heap Sort

Bubble Sort

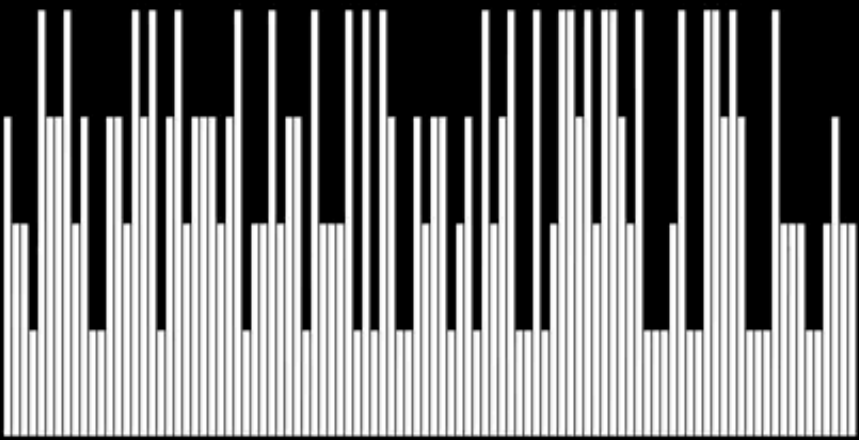
Comb Sort

Cocktail Sort

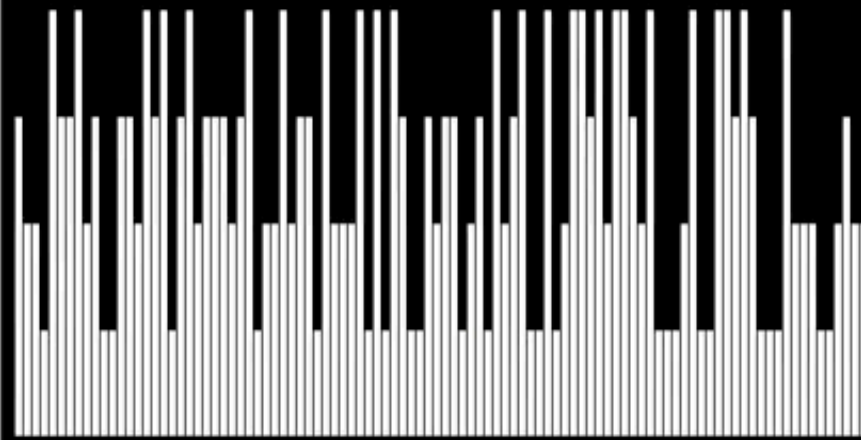


Forces et faiblesses - Few unique

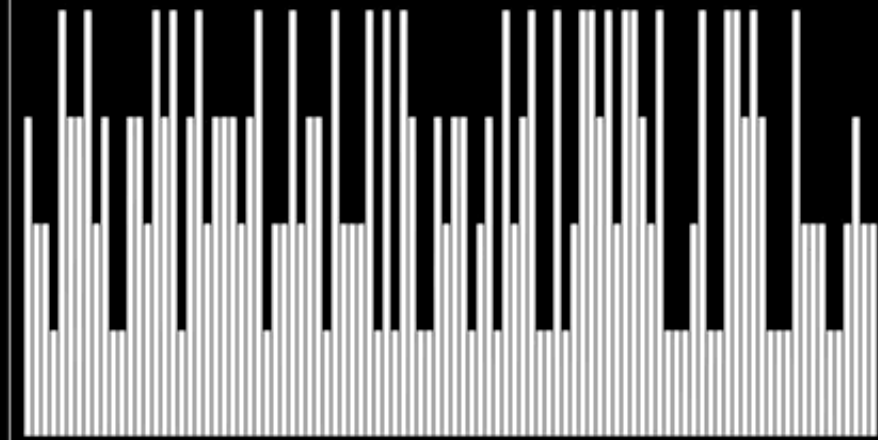
Selection Sort



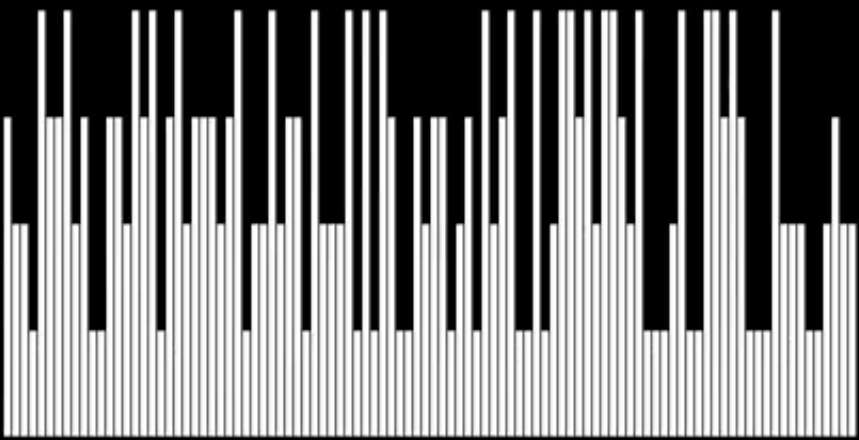
Shell Sort



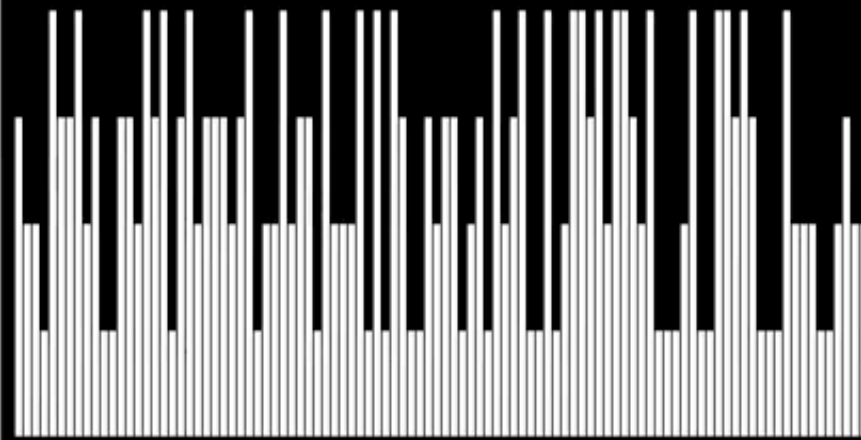
Insertion Sort



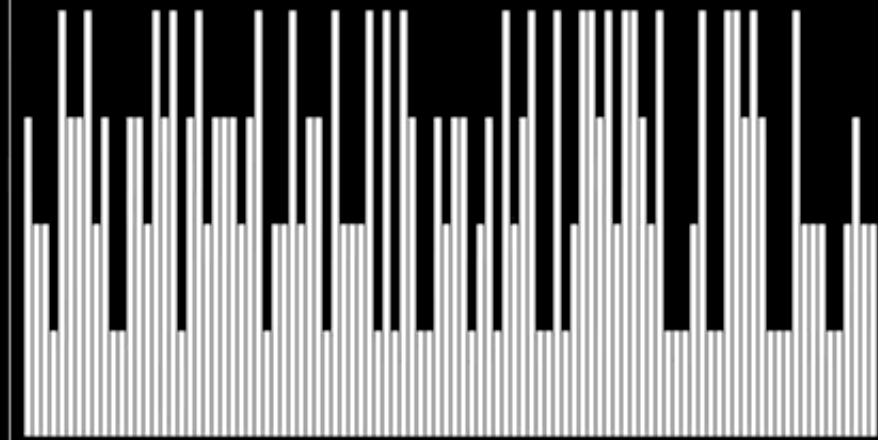
Merge Sort



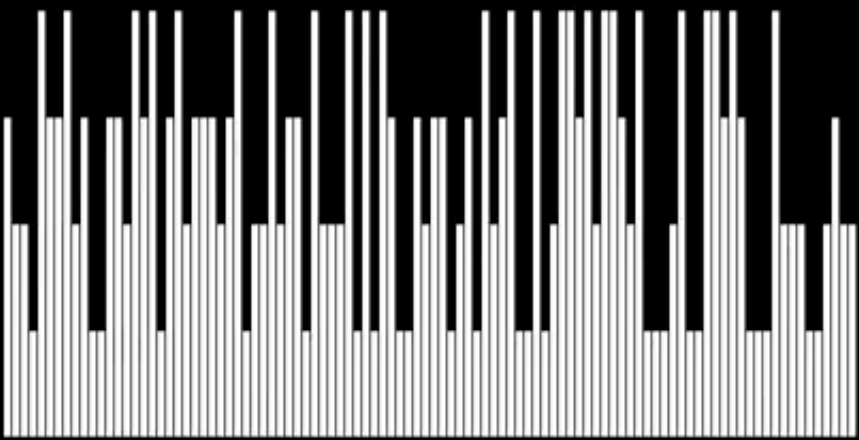
Quick Sort



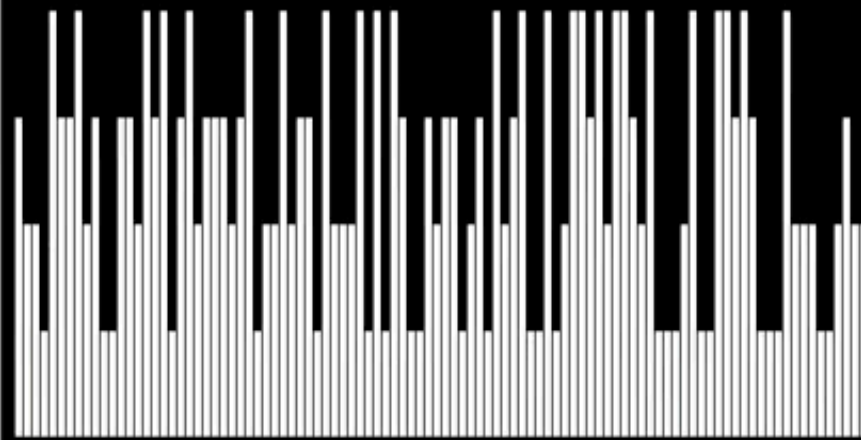
Heap Sort



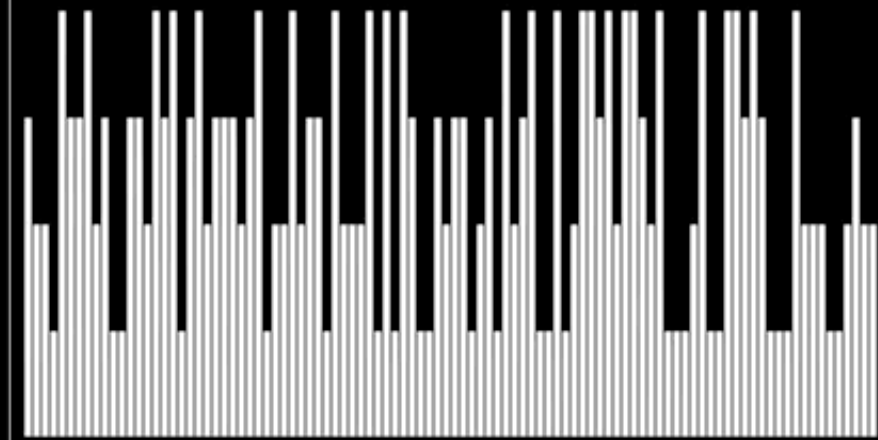
Bubble Sort



Comb Sort



Cocktail Sort



Forces et faiblesses - Reversed

Selection Sort

Shell Sort

Insertion Sort

Merge Sort

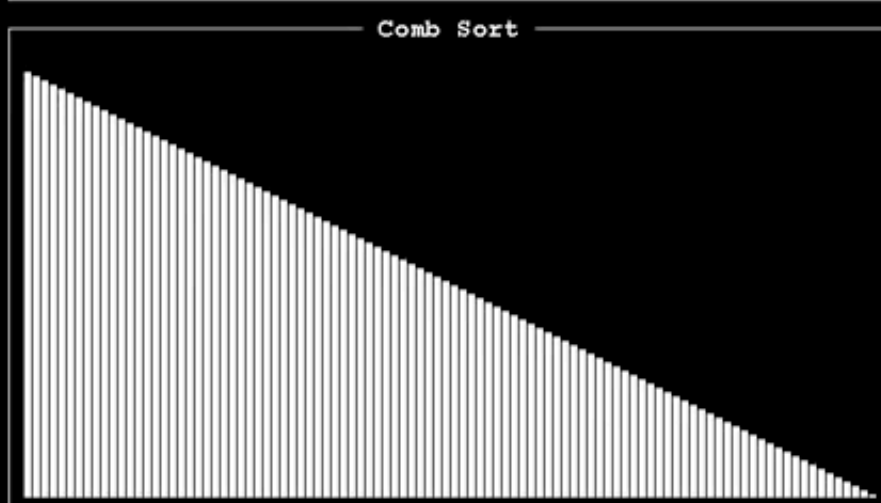
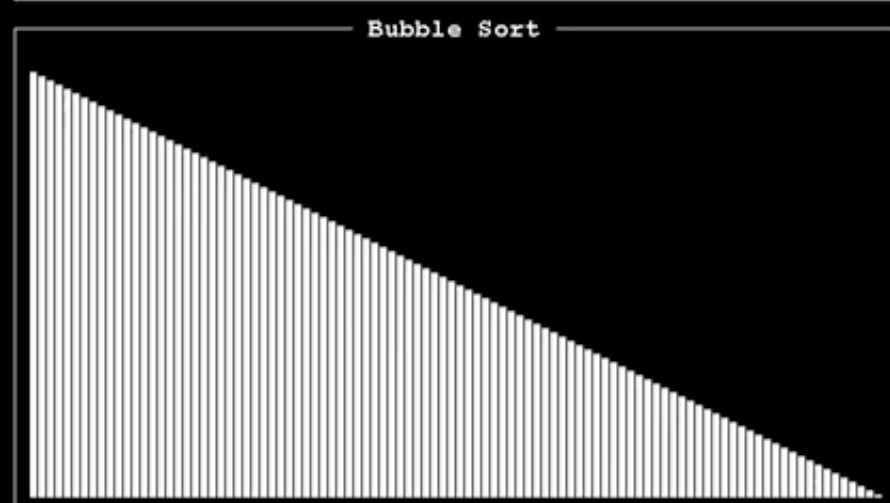
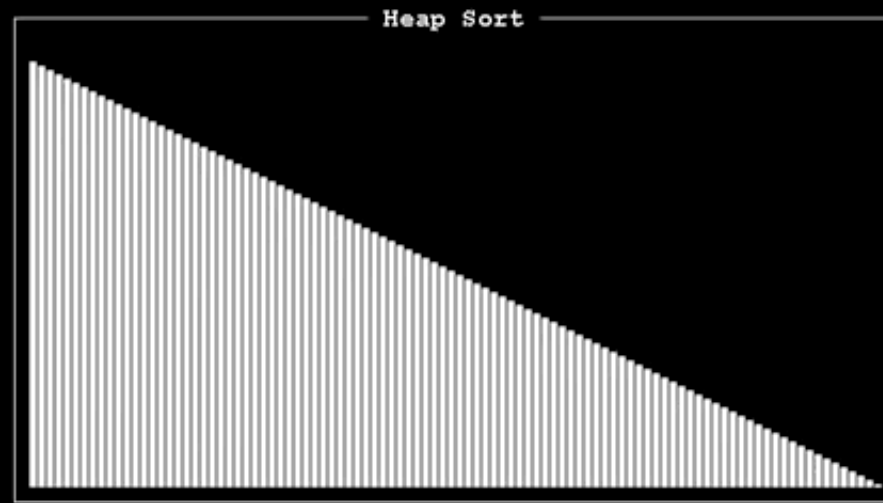
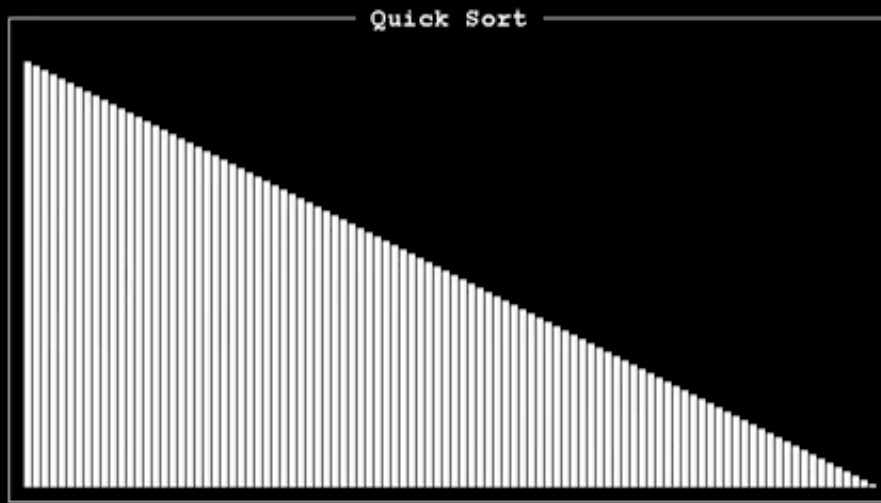
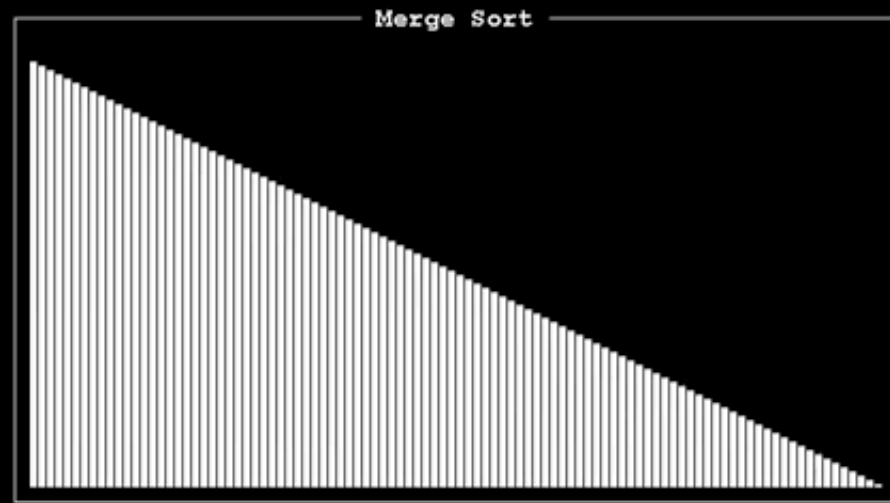
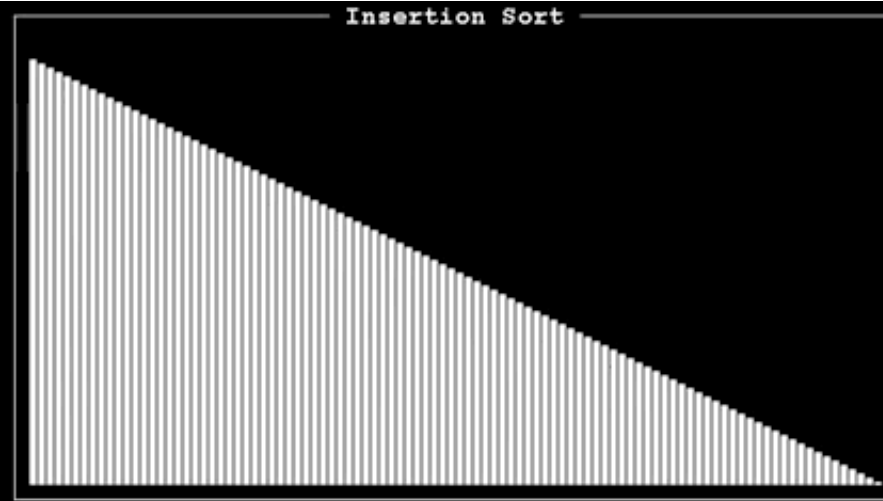
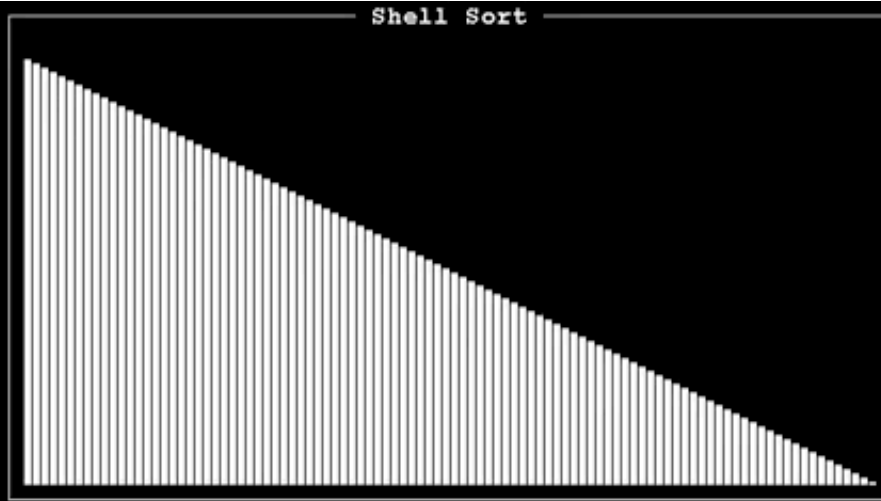
Quick Sort

Heap Sort

Bubble Sort

Comb Sort

Cocktail Sort



Forces et faiblesses - Almost sorted

Selection Sort

Shell Sort

Insertion Sort

Merge Sort

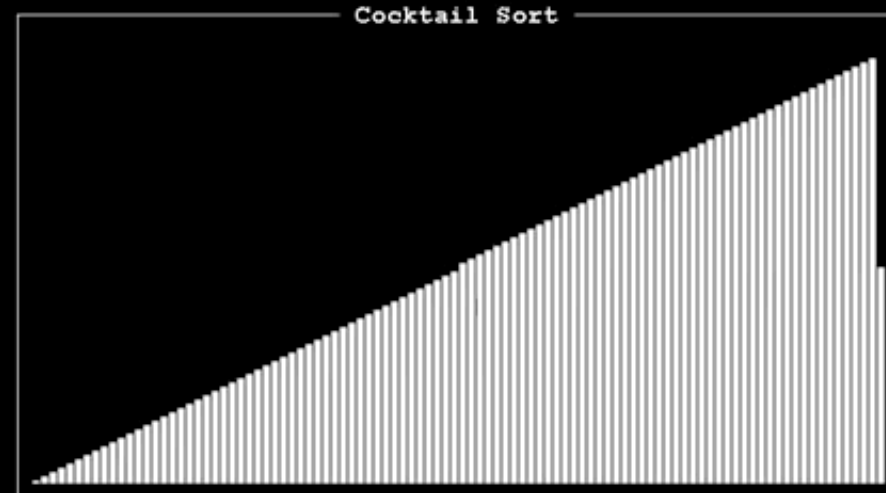
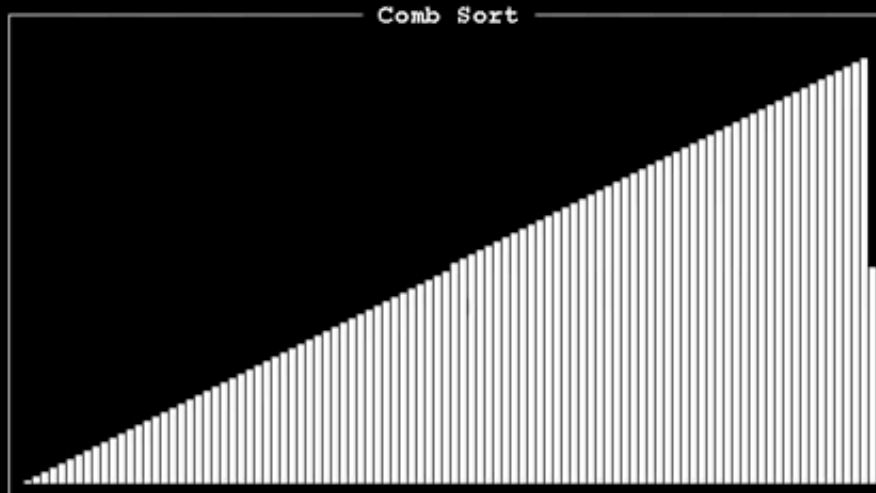
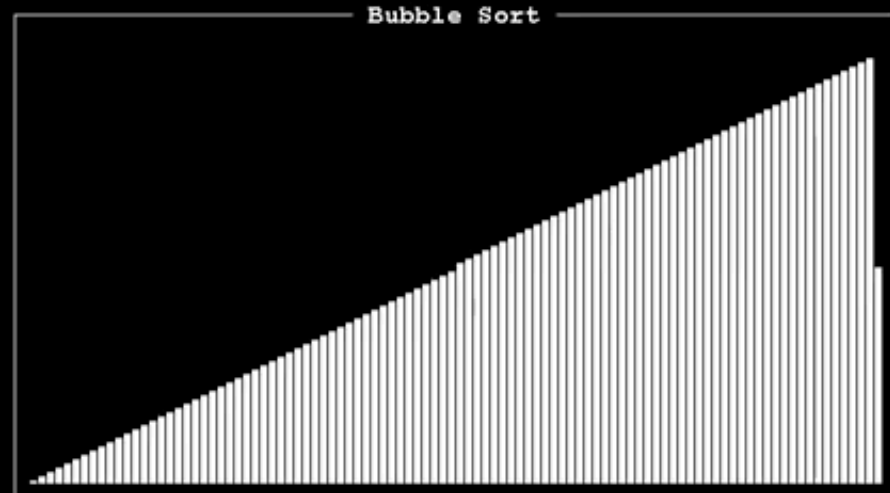
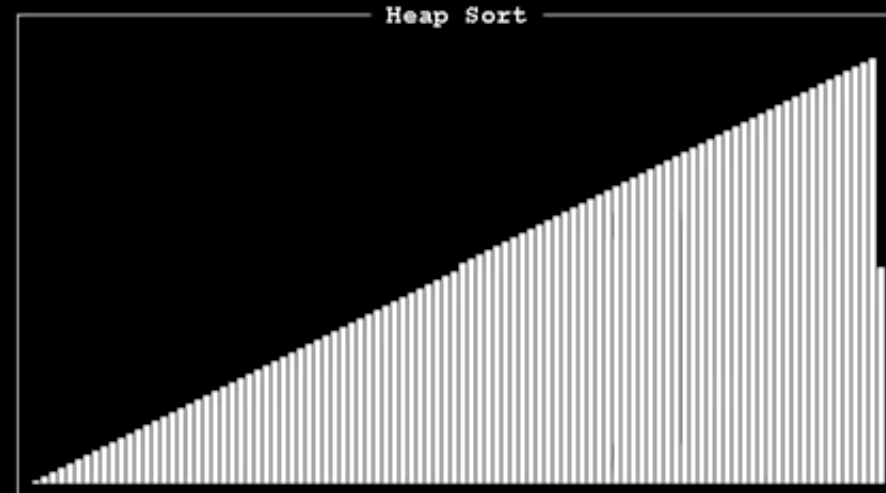
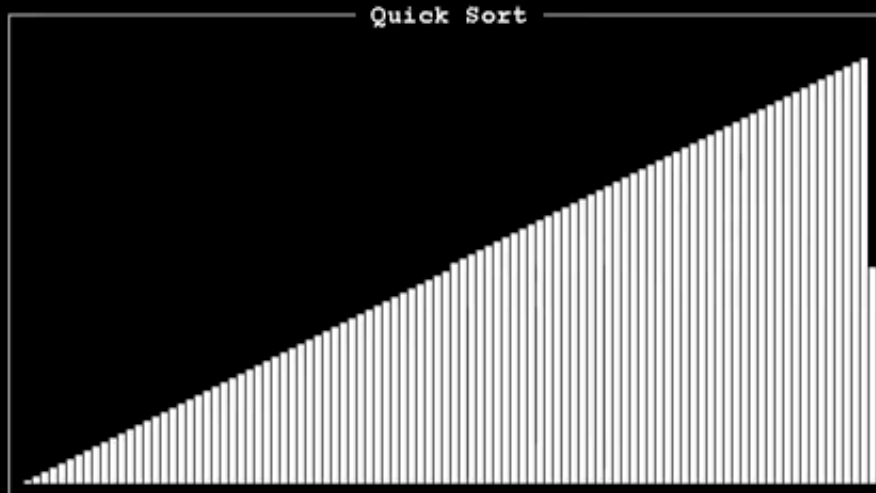
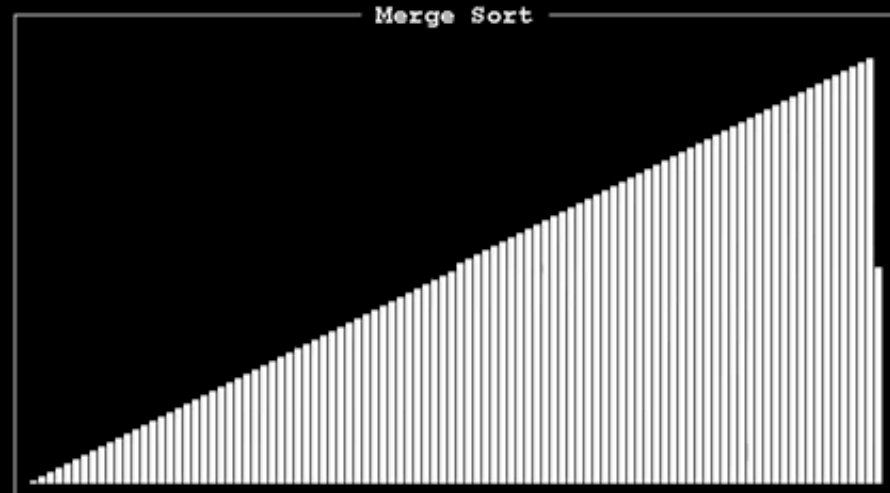
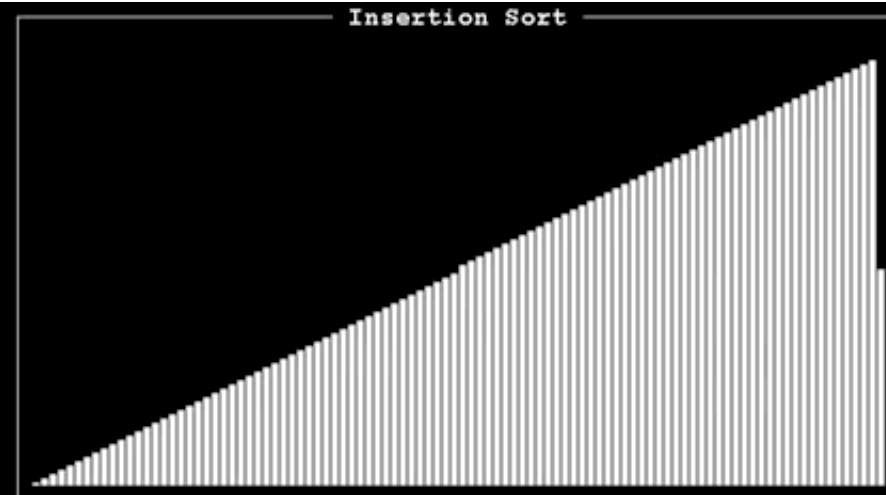
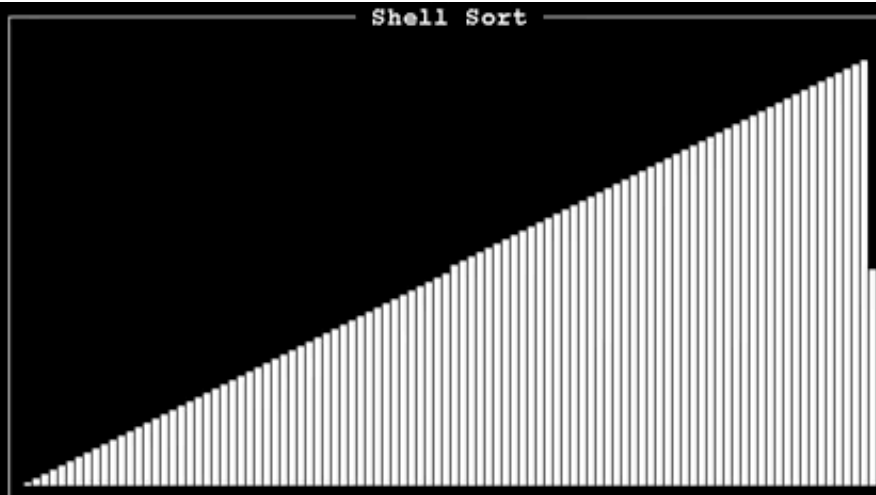
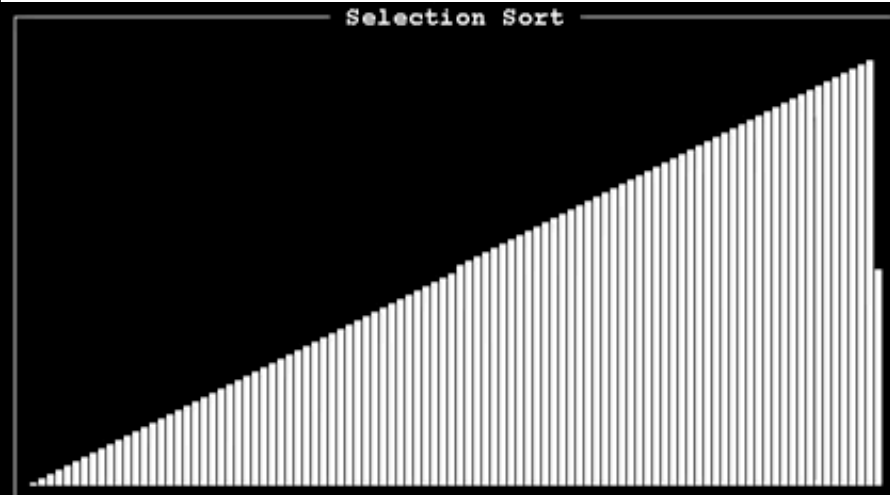
Quick Sort

Heap Sort

Bubble Sort

Comb Sort

Cocktail Sort



QUESTION 2.1.2

n jobs a faire, chaque job prends un temps t_j a être accompli.

$$\min \sum_{j=1} (n - j)t_j$$

Par l'absurde. Soit une suite de jobs optimale telle que $t_a > t_b$ et a est fait immédiatement avant b.

$$2 \cdot t_a + t_b > 2 \cdot t_b + t_a$$

Dès lors, si on inverse a et b, on réduit le temps total. D'où la contradiction.

Vu qu'il ne peut y avoir une paire de jobs mal triée, le tableau doit donc être trié.

QUESTION 2.1.4 TRI DE CARTES

QUESTION 2.1.5 TRI D'UNE LISTE DOUBLEMENT CHÂÎNÉE

QUESTION 2.1.6 NOMBRE DE PAIRES MAL TRIÉES

QUESTION 2.1.7 COMPARABLE/COMPARATOR

QUESTION 2.1.8 RENDRE UN TRI INSTABLE STABLE

QUESTION 2.1.9 TROUVER LA TROISIÈME PLUS GROSSE VALEUR

QUESTION 2.1.10 TROUVER LA NIÈME PLUS GROSSE VALEUR

QUESTION 2.1.11 AUTOBOXING, UNBOXING