Complexité

Michael
XONATIS

Compétence demandée :

Comprendre le vocabulaire de la complexité des algorithmes

1. Etude asymptotique
2. Complexité dans le pire des cas
3. Complexité moyenne
4. Complexité amortie

XONATIS

1. Etude asymptotique
2. Complexité dans le pire des cas
3. Complexité moyenne
4. Complexité amortie

Une étude asymptotique s’effectue lorsque le paramètre d’étude (comme le nombre d’éléments d’un structure de données) tend vers l’infini (c’est-à-dire très grand)

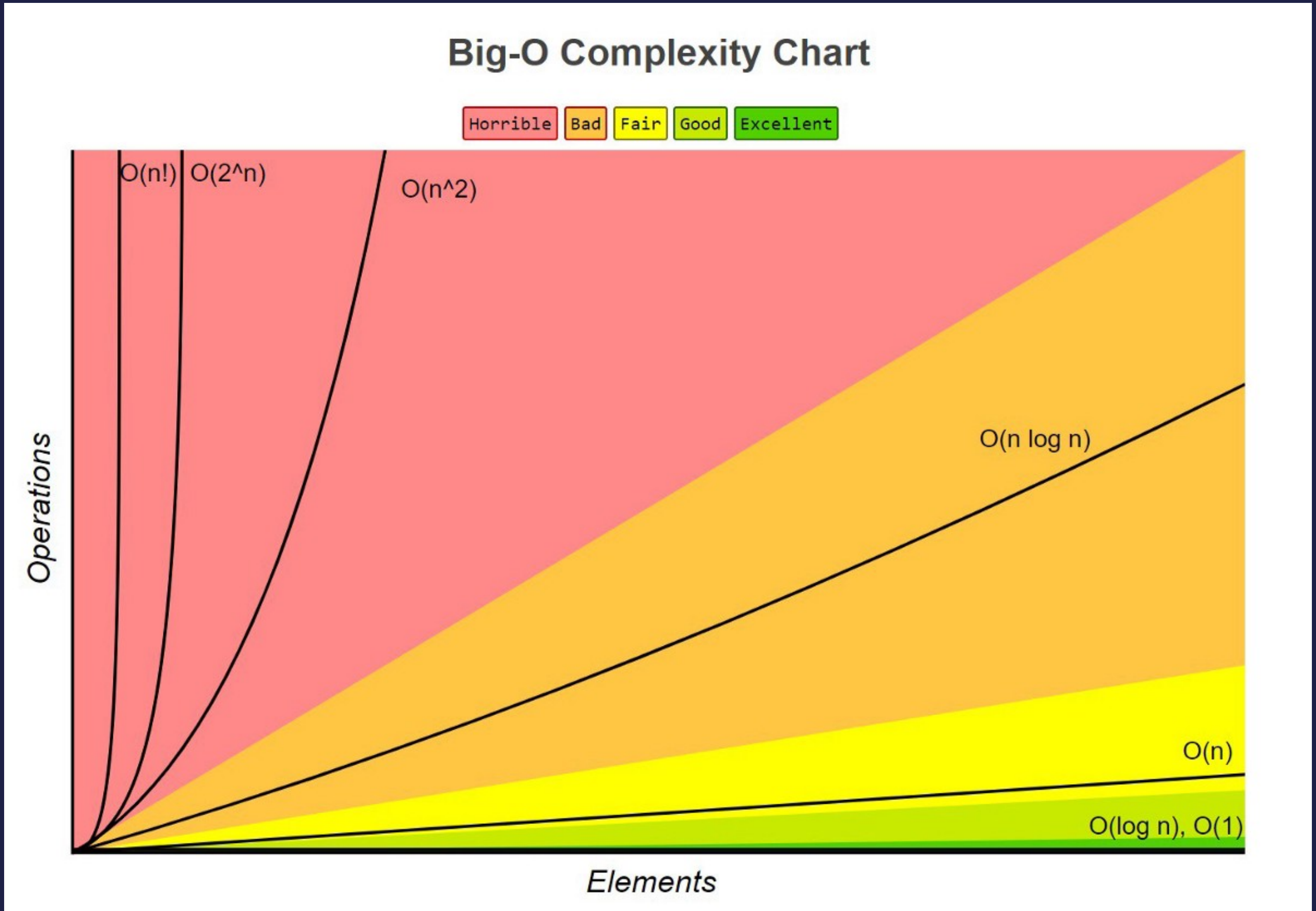
Comportement asymptotique

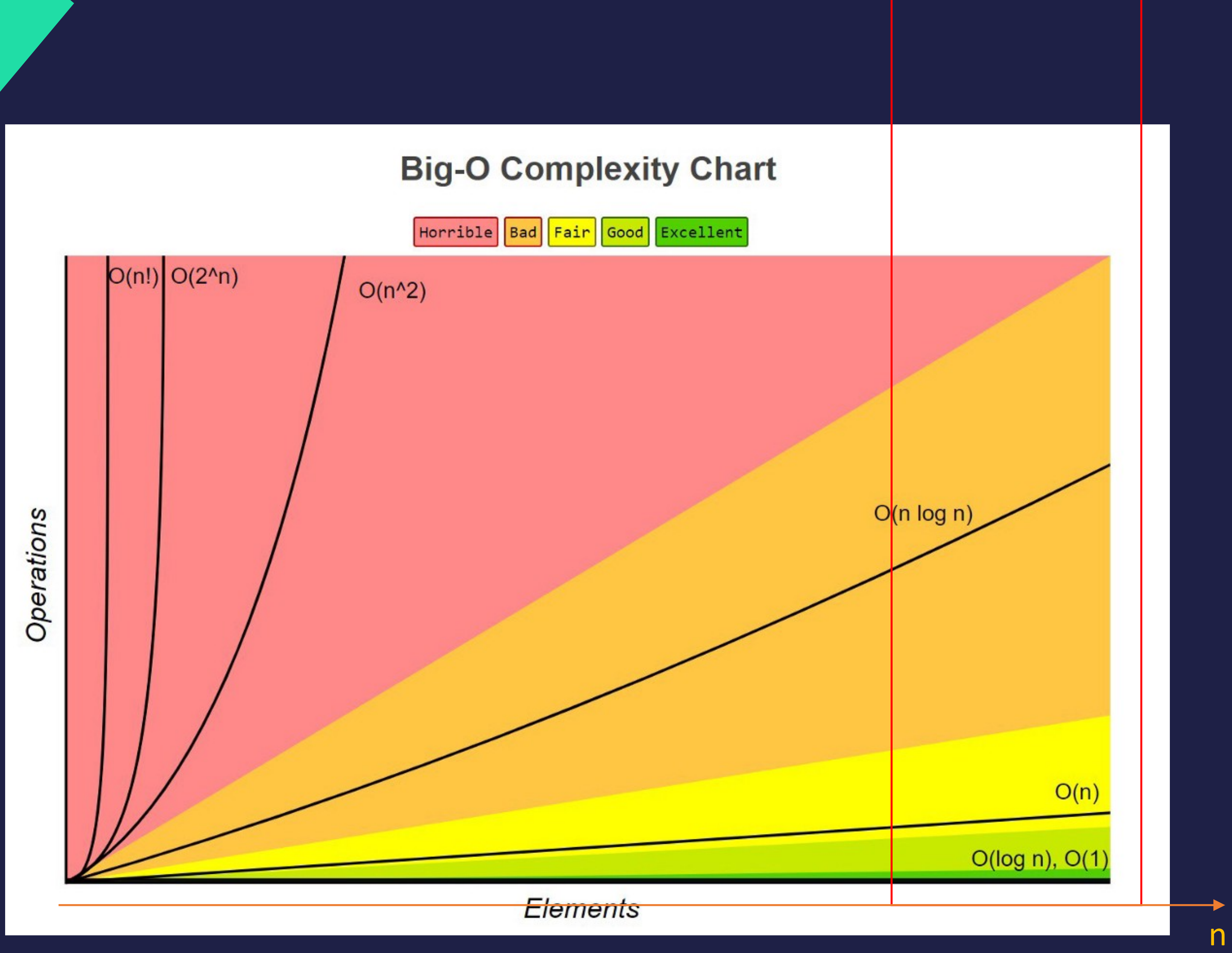
$$O(1) < O(\log n) < O(n) < O(n \log(n)) < O(n*n) < O(\exp(n))$$

Plus rapide

Plus lent

Problème NP-complet


Big-O Complexity Chart
Horrible
Bad
Fair
Good
Excellent
O(n!)
O(2^n)
O(n^2)
O(n log n)
O(n)
O(log n), O(1)
Operations
Elements



Plus n est grand, plus l'écart de temps se creuse entre les algorithmes

Une étude asymptotique s’effectue pour n qui tend vers l’infini

X NATIS

1. Etude asymptotique
2. Complexité dans le pire des cas
3. Complexité moyenne
4. Complexité amortie

En informatique, la complexité dans le pire des cas, mesure la complexité (par exemple en temps ou en espace) d'un algorithme dans le cas le plus défavorable.

Par exemple, si on recherche une valeur dans un tableau (recherche linéaire), la valeur a chercher sera supposée être à la fin du tableau. Pire des cas = pas de bol.

X NATIS

Nom	Cas optimal	Cas moyen	Pire des cas	Complexité spatiale	Stable
Tri rapide	$n \log n$	$n \log n$	n^2	$\log n$ en moyenne, n dans le pire des cas ; variante de Sedgewick : $\log n$ dans le pire des cas	Non
Tri fusion	$n \log n$	$n \log n$	$n \log n$	n	Oui
Tri par tas	$n \log n$	$n \log n$	$n \log n$	1	Non
Tri par insertion	n	n^2	n^2	1	Oui
Introsort	$n \log n$	$n \log n$	$n \log n$	$\log n$	Non
Tri par sélection	n^2	n^2	n^2	1	Non
Timsort	n	$n \log n$	$n \log n$	n	Oui
Tri de Shell	n	$n \log^2 n$ ou $n^{3/2}$	$n \log^2 n$pour la meilleure suite d'espacements connue	1	Non
Tri à bulles	n	n^2	n^2	1	Oui
Tri arborescent	$n \log n$	$n \log n$	$n \log n$ (arbre équilibré)	n	Oui
Smoothsort	n	$n \log n$	$n \log n$	1	Non
Tri cocktail	n	n^2	n^2	1	Oui
Tri à peigne	n	$n \log n$	n^2	1	Non
Tri pair-impair	n	n^2	n^2	1	Oui

XONATIS

1. Etude asymptotique
2. Complexité dans le pire des cas
3. Complexité moyenne
4. Complexité amortie

XONATIS

La complexité en moyenne d'un algorithme mesure la complexité pour une distribution de données en entrée qui a le plus de chances d’arriver.

Par exemple, si on recherche une valeur dans un tableau (recherche linéaire), la valeur a chercher sera supposée être au milieu du tableau.

Nom	Cas optimal	Cas moyen	Pire des cas	Complexité spatiale	Stable
Tri rapide	$n \log n$	$n \log n$	n^2	$\log n$ en moyenne, n dans le pire des cas ; variante de Sedgewick : $\log n$ dans le pire des cas	Non
Tri fusion	$n \log n$	$n \log n$	$n \log n$	n	Oui
Tri par tas	$n \log n$	$n \log n$	$n \log n$	1	Non
Tri par insertion	n	n^2	n^2	1	Oui
Introsort	$n \log n$	$n \log n$	$n \log n$	$\log n$	Non
Tri par sélection	n^2	n^2	n^2	1	Non
Timsort	n	$n \log n$	$n \log n$	n	Oui
Tri de Shell	n	$n \log^2 n$ ou $n^{3/2}$	$n \log^2 n$ pour la meilleure suite d'espacements connue	1	Non
Tri à bulles	n	n^2	n^2	1	Oui
Tri arborescent	$n \log n$	$n \log n$	$n \log n$ (arbre équilibré)	n	Oui
Smoothsort	n	$n \log n$	$n \log n$	1	Non
Tri cocktail	n	n^2	n^2	1	Oui
Tri à peigne	n	$n \log n$	n^2	1	Non
Tri pair-impair	n	n^2	n^2	1	Oui

XONATIS

XONATIS

1. Etude asymptotique
2. Complexité dans le pire des cas
3. Complexité moyenne
4. Complexité amortie

La complexité amortie est la mesure de la complexité dans un but d'efficacité.

La complexité amortie est la moyenne des complexités dans le pire des cas sur plusieurs run, c'est-à-dire sur plusieurs exécutions de l'algorithme.