Ordenação dos dois terços

O algoritmo mais interessante do oeste

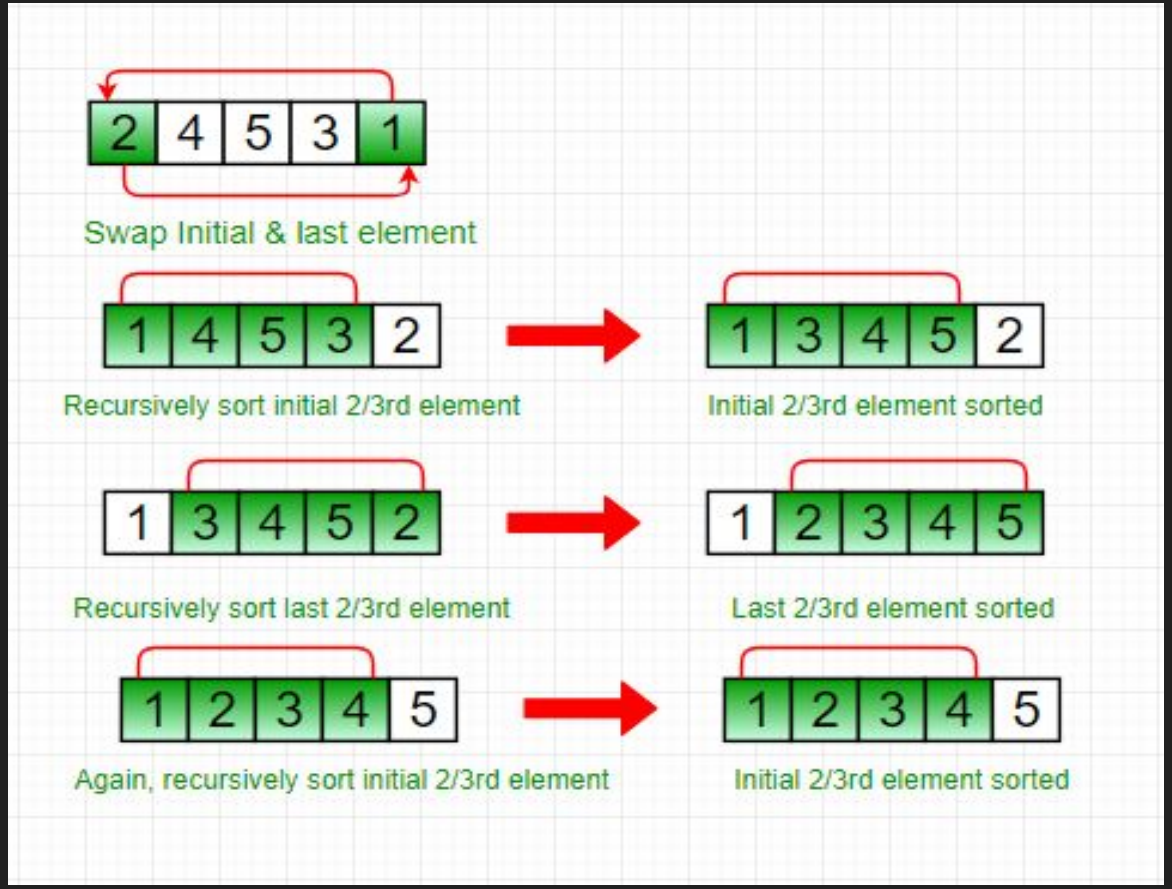
Referências

Michal Forišek, Stooge Sort -

Disponível em <https://www.quora.com/What-algorithms-have-the-most-unexpected-big-O-time-complexity>

Exemplo de Implementação -

Disponível em <https://www.geeksforgeeks.org/stooge-sort/>


2 4 5 3 1
Swap Initial & last element
1 4 5 3 2
Recursively sort initial 2/3rd element
1 3 4 5 2
Initial 2/3rd element sorted
1 3 4 5 2
Recursively sort last 2/3rd element
1 2 3 4 5
Last 2/3rd element sorted
1 2 3 4 5
Again, recursively sort initial 2/3rd element
1 2 3 4 5
Initial 2/3rd element sorted

Complexidade:

Recorrência: $$a(x) = 3a(\frac{2}{3}x) + 1$$

(CLRS3d) Para n inteiro, seja uma recorrência DC $a(n)$ com custo de combinar $f(n)$:

$$a(n) = \alpha a(n/\beta) + f(n)$$

(caso 1) se	$f(n) = O(n^{\log_\beta \alpha - \epsilon})$ e $\epsilon > 0$	$a(n) = \Theta(n^{\log_\beta \alpha})$
(caso 2) se	$f(n) = \Theta(n^{\log_\beta \alpha})$	$a(n) = \Theta(n^{\log_\beta \alpha} \lg n)$
(caso 3) se	$f(n) = \Omega(n^{\log_\beta \alpha + \epsilon})$ e $\epsilon > 0$	$a(n) = \Theta(f(n))$

Complexidade

(caso 1) se $f(n) = O(n^{\log_\beta \alpha - \epsilon})$ e $\epsilon > 0$ $a(n) = \Theta(n^{\log_\beta \alpha})$

$a(n) = 3.a(\frac{2}{3}n) + 1$

$\alpha = 3$

$\beta = \frac{3}{2}$

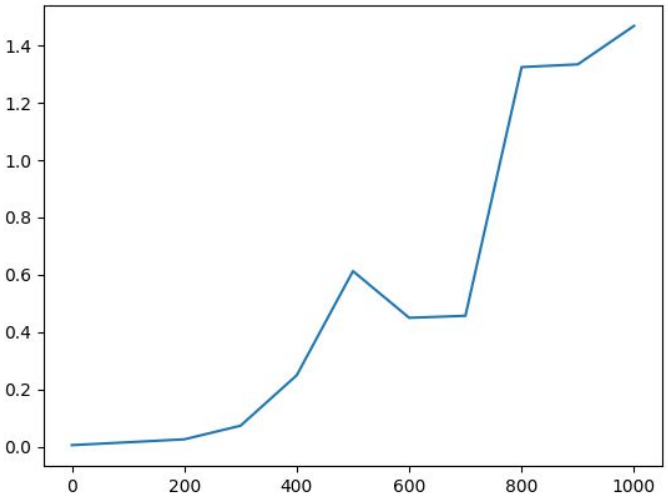
$f(n) = 1$

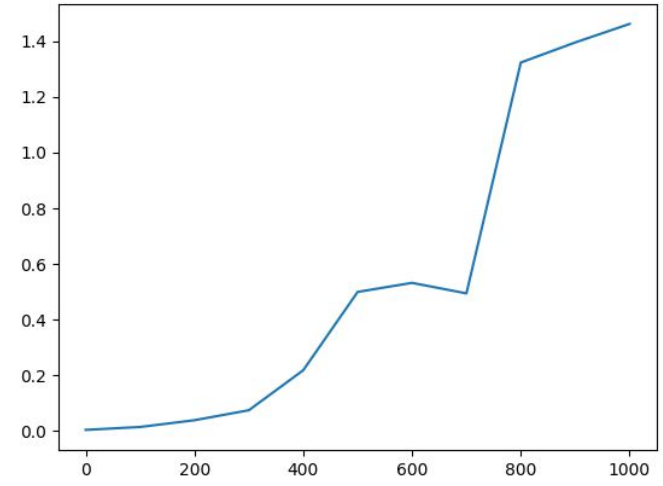
$$f(n) = O(n^{\log_{\frac{3}{2}} 3 - e})$$

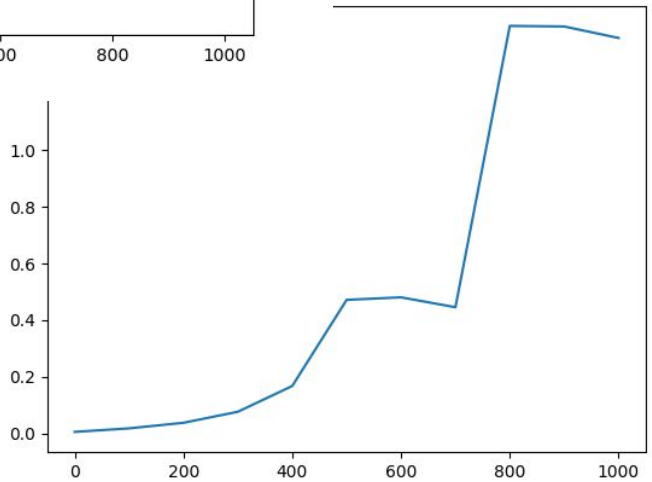
escolhendo $e = 2$,

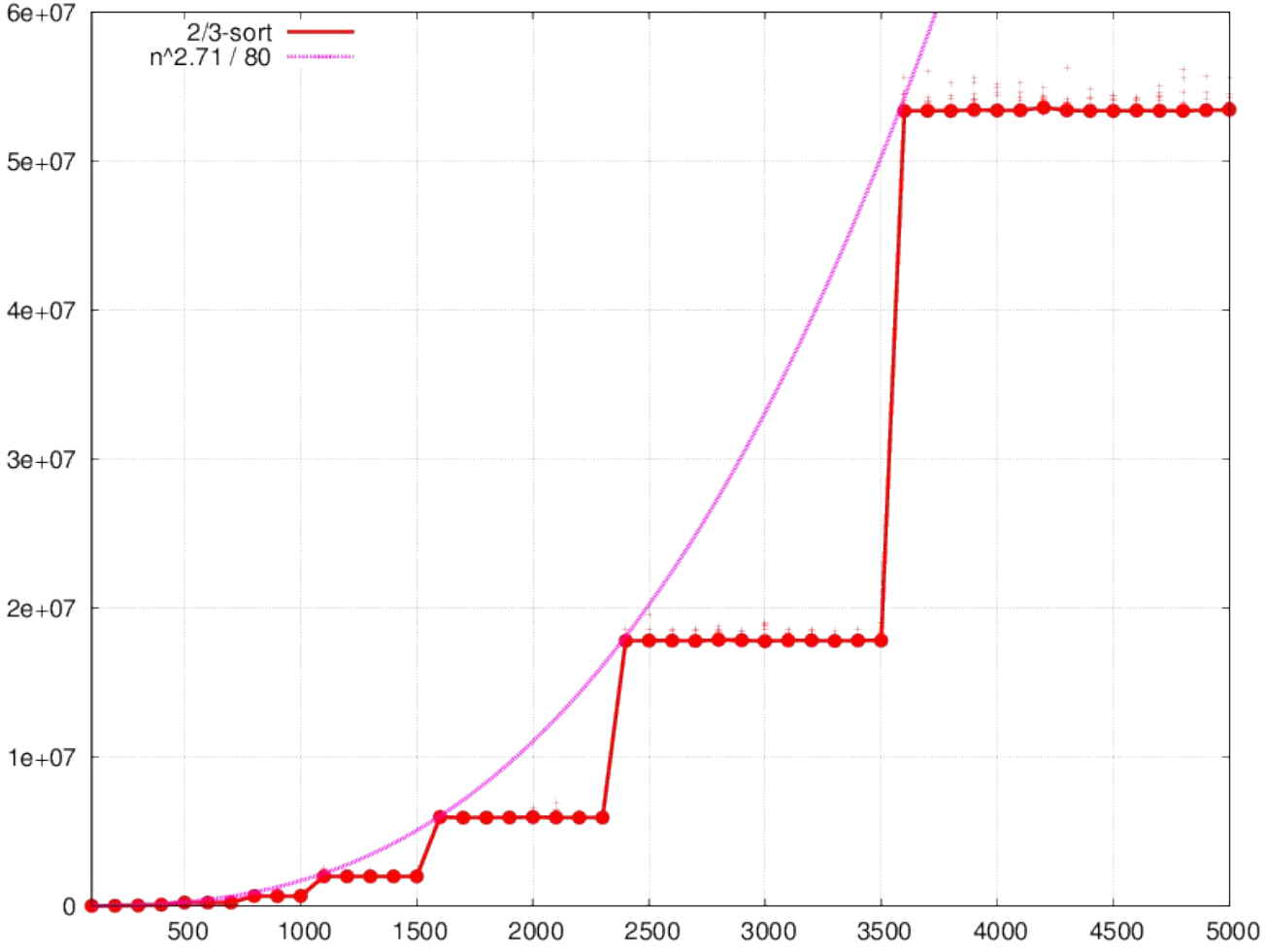
$$f(n) = O(n^{\log_{\frac{3}{2}} 1}) = O(n^{\log_{\frac{3}{2}} 1}) = O(n^0) = O(1)$$

$$a(n) = \Theta(n^{\log_{\frac{3}{2}} 3}) = \Theta(n^{2.7095})$$


1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
200
400
600
800
1000


1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
200
400
600
800
1000


1.0
0.8
0.6
0.4
0.2
0.0
0
200
400
600
800
1000


2/3-sort
n^2.71 / 80
6e+07
5e+07
4e+07
3e+07
2e+07
1e+07
0
500
1000
1500
2000
2500
3000
3500
4000
4500
5000

Variação

$$a(n) = 2.a(\frac{2}{3}n) + \frac{2}{3}n$$

(caso 1) se $f(n) = O(n^{\log_\beta \alpha - \epsilon})$ e $\epsilon > 0$ $a(n) = \Theta(n^{\log_\beta \alpha})$

$$f(n) = O(n^{log_{\frac{3}{2}} 3-e}) = O(n^{log_{\frac{3}{2}} \frac{3}{2}}) = O(n)$$

$$a(n) = \Theta(n^{log_{\frac{3}{2}} 2}) = \Theta(n^{1.7095})$$