



• U •

FCTUC FACULDADE DE CIÊNCIAS
E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Faculdade de Ciências e Tecnologias – Universidade de Coimbra
Departamento de Engenharia Informática
Mestrado em Engenharia Informática

Metodologias Experimentais em Informática

2019/2020

Diogo Albuquerque

nº 2015248499

diogoma@student.dei.uc.pt

Renato Carvalho

nº 2015235034

renatomsc@student.dei.uc.pt

Renato Santos

nº 2015237457

renatojms@student.dei.uc.pt

Introdução

O objetivo deste trabalho será tentar responder à pergunta “Como os algoritmos de ordenação são afetados por falhas de memória?”.

Para isso é preciso ter em conta todas as variáveis que podem influenciar os resultados da experiência de modo a ter um melhor entendimento sobre a questão que se está a estudar.

Existem dois tipos de variáveis. As variáveis dependentes e as variáveis independentes. As variáveis dependentes são aquelas em que o valor depende das variáveis independentes. As variáveis independentes são as que representam uma grandeza a ser alterada.

As variáveis independentes que serão analisadas neste trabalho são: o número de números a serem ordenados, ou, por outras palavras, o tamanho de array de input a ser ordenado (N), a probabilidade de ocorrer um erro na geração dos números de input (EPS) e o intervalo máximo possível dos números gerados para input ($MaxR$).

As variáveis dependentes analisadas serão o tamanho da maior sequência ordenada e o número de falhas na comparação no decorrer da execução de cada algoritmo.

Iremos usar como algoritmos de análise os algoritmos bubblesort, quicksort, mergesort e insertionsort.

Procedimento

Para retirar algum tipo de conclusão minimamente válida era necessário obter uma quantidade de dados considerável, portanto, para cada input gerado (com um conjunto de condições) foram executados mil e quinhentas vezes cada algoritmo de ordenamento.

A variável N variou entre os valores: 1000, 3000, 6000, 8000 e 10000; o EPS variou entre $1/N$, $5/N$ e $10/N$ e o MaxR variou entre $N/2$ e N.

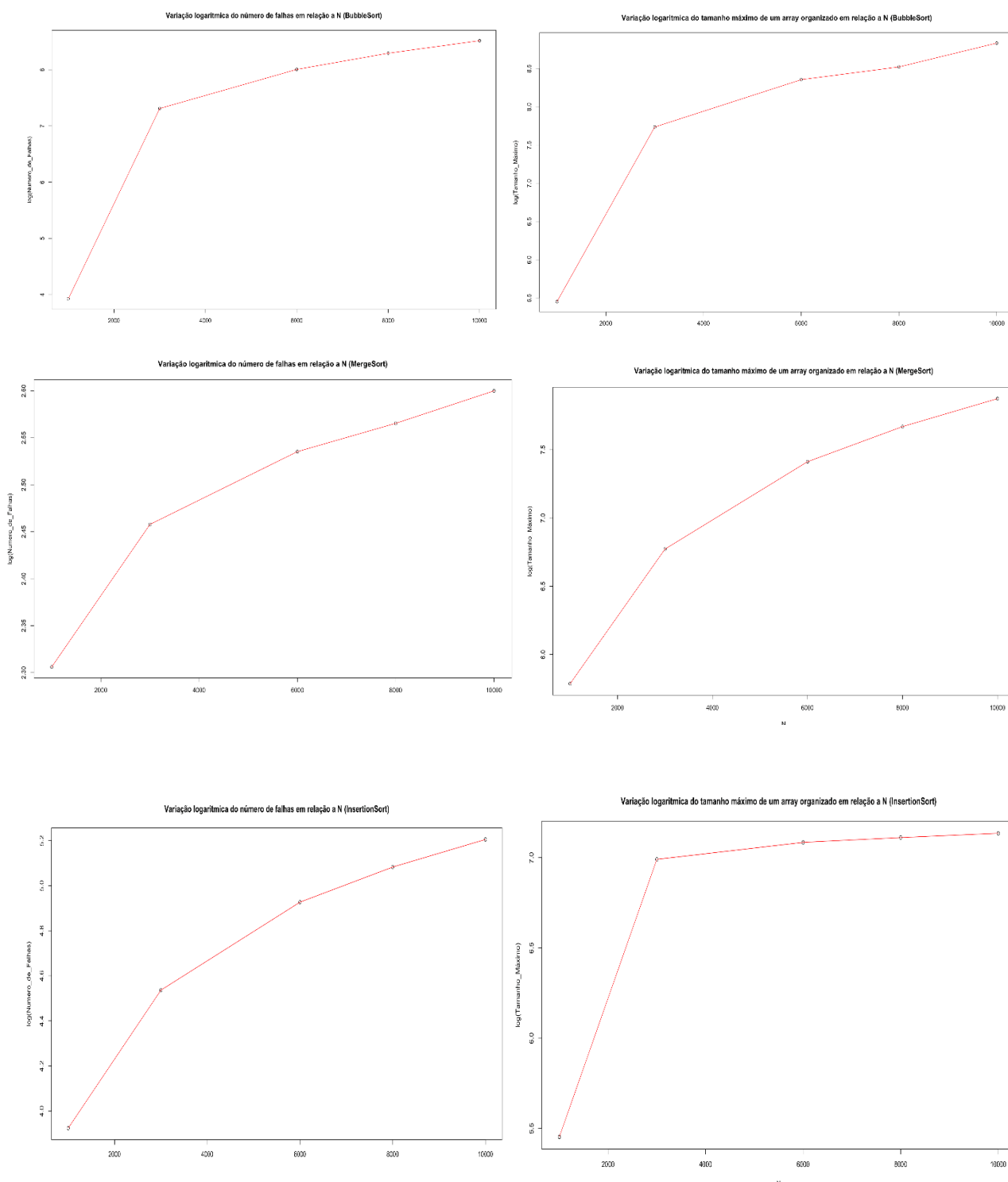
Cada ficheiro de output continha, por ordem, o valor de EPS e de N utilizados bem como o tamanho da maior sequência ordenada e o número de falhas na comparação no decorrer da execução de cada algoritmo.

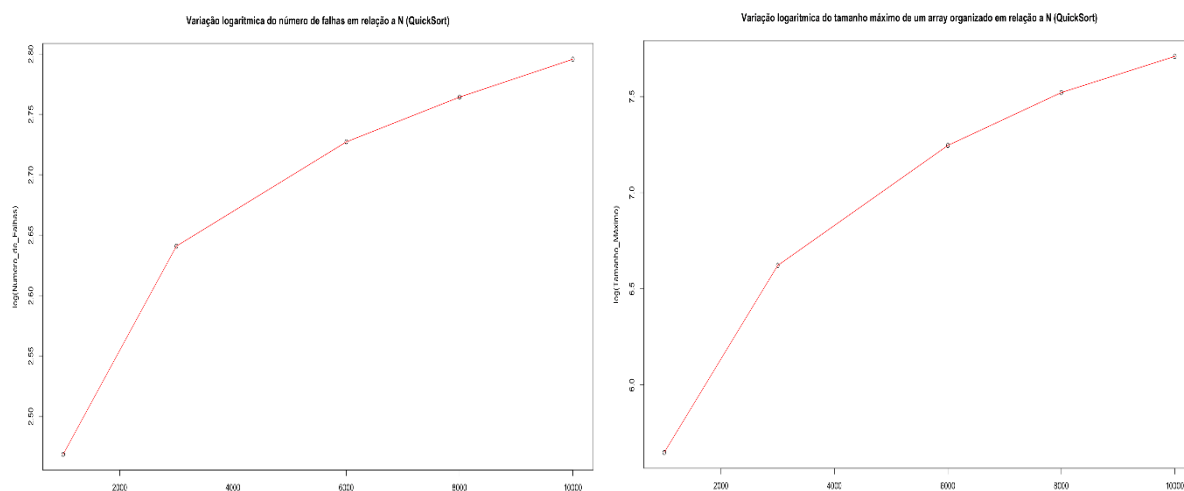
Esses ficheiros foram transformados e enviados para serem analisados com recurso ao RStudio (linguagem R) sendo organizados e dispostos em gráficos para melhor entendimento do comportamento e resposta dos algoritmos de ordenação. Gráficos esses que serão expostos e comentados a seguir.

Resultados obtidos

Os valores utilizados na construção dos gráficos são a media dos valores obtidos para cada caso e recorremos a uma visão logarítmica para o melhor visionamento dos resultados

- Variação do N (com $EPS=1/N$ e $MaxR=N/2$):**

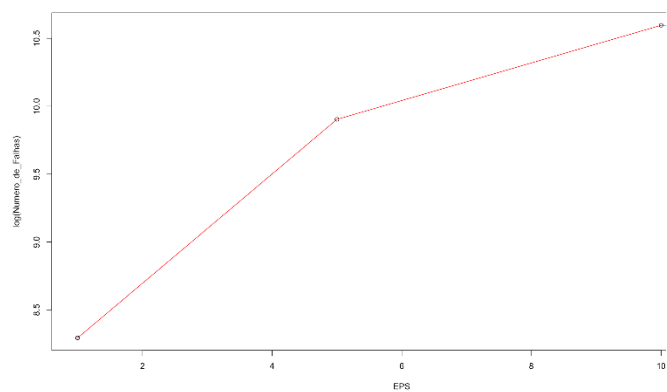




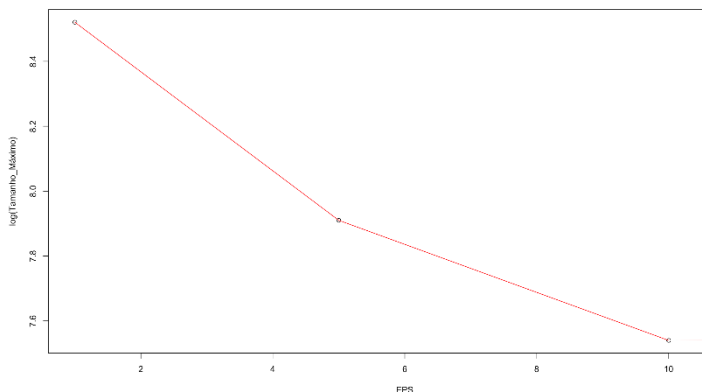
De um modo geral é possível verificar uma relação logarítmica entre o aumento do número de falhas e o aumento do N e entre o aumento do tamanho máximo de um array e o do N. De notar apenas no caso do Insertionsort no que diz respeito à análise ao tamanho máximo do array existe um crescimento mais acentuada seguido de uma estabilização o que contrasta um pouco com o observado nos restantes resultados.

• **Variação do EPS (com $N=8000$ e $\text{MaxR}=N/2$):**

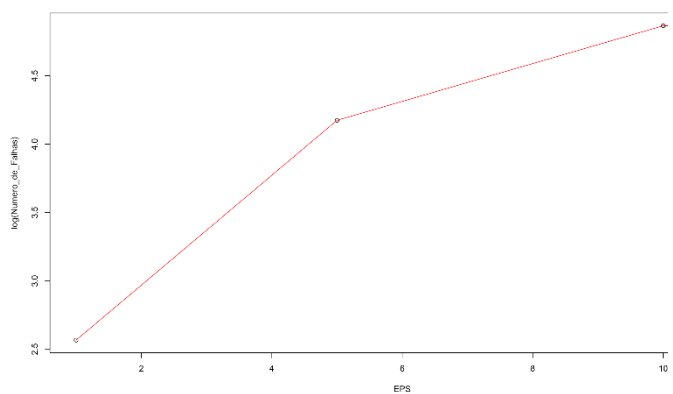
Variação logarítmica do número de falhas em relação a N (BubbleSort)



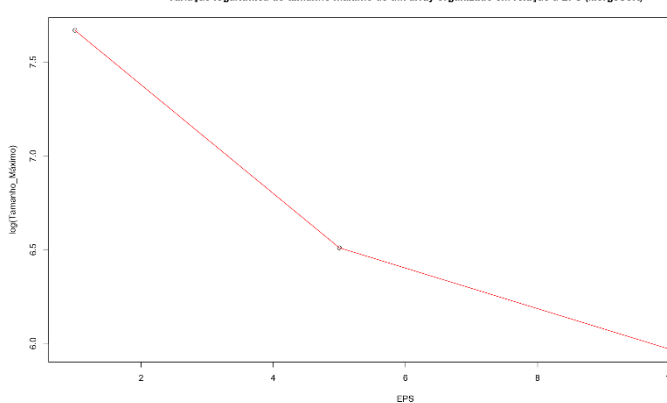
Variação logarítmica do tamanho máximo de um array organizado em relação a EPS (BubbleSort)



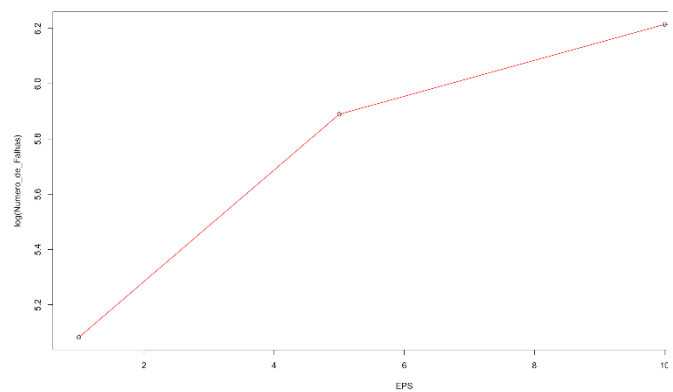
Variação logarítmica do número de falhas em relação a N (MergeSort)



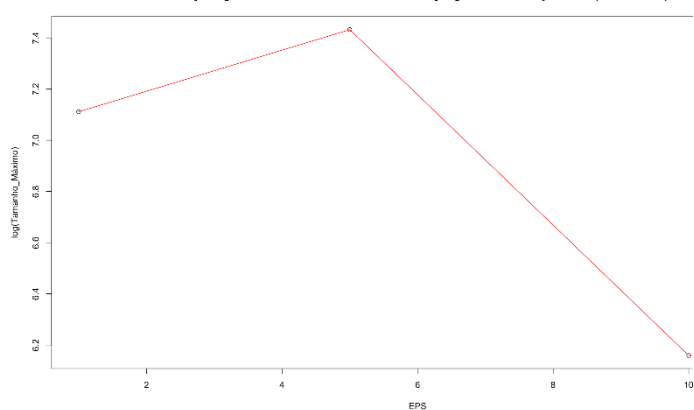
Variação logarítmica do tamanho máximo de um array organizado em relação a EPS (MergeSort)



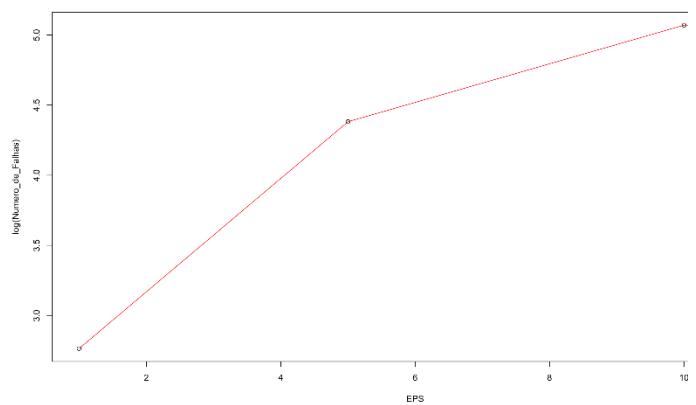
Variação logarítmica do número de falhas em relação a EPS (InsertionSort)



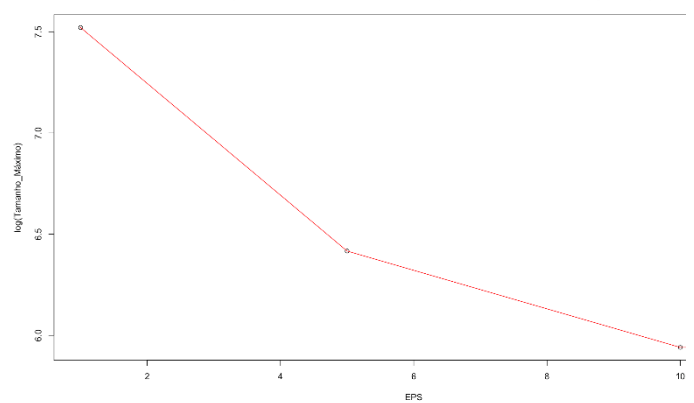
Variação logarítmica do tamanho máximo de um array organizado em relação a EPS (InsertionSort)



Variação logarítmica do número de falhas em relação a EPS (QuickSort)



Variação logarítmica do tamanho máximo de um array organizado em relação a EPS (QuickSort)



Ao contrário do que aconteceu com o a variação do N existe uma diferença clara na tendência de crescimento dos valores do número de falhas nos algoritmos e o crescimento do tamanho máximo de um array. Enquanto que a primeira tendência é claramente positiva a segunda é negativa (com a exceção no caso do Insertionsort).

- **Variação do MaxR (com $N=8000$ e $EPS=1/N$):**

No caso da variação do intervalo máximo possível dos números gerados para input o que foi observado não suficientemente significativo para ser alvo de reflexão, uma vez que, as oscilações a nível das médias testadas foram muito reduzidas. Por essa razão não apresentamos os gráficos apesar de seguirem em anexo.

Conclusão:

Analisando a variação do tamanho do array de input (N) a ser ordenado podemos concluir que, com o seu aumento, o número de falhas e o tamanho máximo do array ordenado crescem de forma logarítmica. Algo que era esperado pois, com o aumento do valor de N , o valor da probabilidade de ocorrência de falha diminui para valores próximos de zero, o que leva a que o número de falhas estabilize e se torne cada vez mais próxima de zero.

Em relação à variação da probabilidade de ocorrer um erro na geração dos números de input, podemos concluir que, com o seu aumento, o número de falhas aumenta de forma linear e o tamanho máximo do array diminui. Também isto é expectável uma vez que, com o aumento da probabilidade, as falhas irão (teoricamente) aumentar em frequência e consequentemente diminuir o tamanho máximo do array ordenado.

Dos quatro algoritmos abordados alguns comportam-se de forma muito semelhante nas diferentes situações, porém o InsertionSort é mais sensível a falhas de memória pois é o que apresenta maiores variações dos testes efetuados.