

Relatório Projeto 3.4 AED 2021/2022

Nome: Mariana Lopes Paulino

Nº Estudante: 2020190448

TP (inscrição): 1

Login no Mooshak: 2020190448

Registrar os tempos computacionais das variantes exploradas nos subprojetos anteriores para os dois cenários. Os tamanhos das sequências (N) devem ser: 10000, 100000, 1000000, 10000000, 100000000. Só devem ser contabilizados os tempos das operações de inserção e pesquisa. Exclui-se o tempo de leitura do input e de impressão dos resultados. Não devem apresentar regressões.

Tabela para o cenário 1

N	PESQUISA LINEAR	PESQUISA BINÁRIA	ÁRVORE BINÁRIA	ÁRVORE AVL
0	9.54E-07	5.01E-06	5.01E-06	7.15E-07
5000	1.84495616	1.543895721	0.118971825	0.17730999
10000	19.64517307	14.69772577	0.391508818	0.407347918
15000	96.43667102	81.14184999	0.875909805	0.795706987

Tabela para o cenário 2

N	PESQUISA LINEAR	PESQUISA BINÁRIA	ÁRVORE BINÁRIA	ÁRVORE AVL
0	9.54E-07	5.01E-06	5.01E-06	7.15E-07
5000	0.230208874	0.024022579	0.015013933	0.017016172
10000	1.946769476	0.131123304	0.046042204	0.048044132
15000	7.942814112	0.464422464	0.105095148	0.104094267

Gráfico para o cenário 1

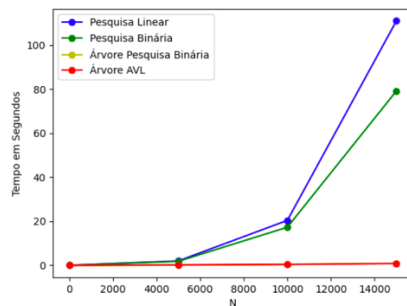
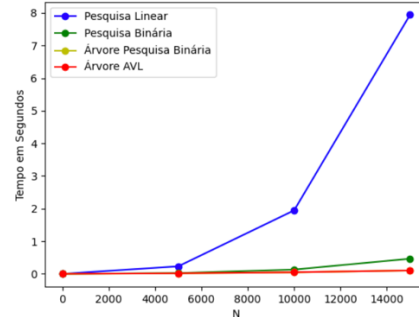


Gráfico para o cenário 2



Análise dos resultados e, adicionalmente, discussão sobre como se comportariam estas variantes na operação de remover um elemento do conjunto:

Após uma grande alteração dos N analisados, [0, 50 000] obtemos os resultados mais expectáveis possível, uma vez que as árvores binária e AVL serão muito melhores que os métodos de pesquisa linear ou binária, sendo a pesquisa linear a pior de todas nos dois cenários uma vez que todas as outras acabam sempre por estar mais organizadas tanto para consulta como para inserção uma vez que utilizamos o número de utente para localizar o local tanto para consultar como para inserir um novo caso, sendo este o único método onde temos de percorrer todo o array para encontrar o sitio ou não no caso de não existir para confirmar se existe ou não já o registo. No cenário 1 foca 90% dos seus dados em inserção pelo que esta vai ser mais lenta que

apenas a consulta porque na inserção para além de procurar pelo sítio onde inserir os dados temos também de os inserir e acabar por atualizar a nova informação.

A eliminação de um elemento da árvore ou lista organizada de pesquisa acaba por ser similar à inserção de um elemento uma vez que quando inserimos um elemento temos de garantir que todos os elementos do array à direita têm de “abrir” espaço para o novo elemento, por exemplo o elemento x vai ser inserido entre i e j uma vez que se encaixa no meio dos mesmos, o elemento i mantém a sua posição no array, o x irá ocupar a posição de j e j irá assumir a posição em que estava $+1$. Agora na eliminação seguindo a mesma linha de pensamento se tivermos os elementos considerados anteriormente i , x , j e pretendemos agora eliminar o elemento i , ambos x e j irão não aumentar um índice mas sim subtrair um, ou seja, todos os elementos à direita daquele que é para ser retirado terão de ter um índice subtraído à sua posição.