

Aluno: Jhonatan Guilherme de Oliveira Cunha

RA: 2135590

Disciplina: Algoritmos e Estrutura de Dados 2

VETORES DECRESCENTES

	N			
	1000	10000	100000	500000
ABB	0,01500	0,32800	40,28200	982,876
ARN	0,00300	0,03100	0,48300	3,75800
AVL	0,00000	0,64100	3,09300	3,56300

Figure 1: Tempo de Execução (em s) para Ordenar Vetores com N Elementos em Ordem Crescente.

	N			
	1000	10000	100000	500000
ABB	999	9999	99999	499999
ARN	9	13	16	18
AVL	9	13	16	18

Figure 2: Altura das Árvores Antes do Percurso Em-Ordem.

VETORES ALEATÓRIOS

	N			
	1000	10000	100000	500000
ABB	0.00150	0,00310	0,03440	0,15510
ARN	0.00160	0,00780	0,07340	0,34690
AVL	0.00000	0,00310	0,01410	0,05780

Figure 3: Tempo de Execução (em s, média +- desvio) para Ordenar Vetores Aleatórios com N Elementos.

	N			
	1000	10000	100000	500000
ABB	642,2	5570,3	19697,6	23349,8
ARN	8	10,8	12,2	12,4
AVL	11	14,8	17	17

Figure 4: Media Altura das Árvores Antes do Percurso Em-Ordem.

f. Houve diferença significativa entre o tempo de execução entre os três métodos avaliados no caso da ordenação dos vetores inicialmente em ordem decrescente? Se sim, qual foi o melhor método? Qual foi o pior? Justifique sua resposta.

A princípio quando o valor de $N \leq 10000$, a diferença de tempo de execução entre os três métodos não é algo tão gritante. Porém, quando o valor é maior começa aparecer diferenças bem notáveis e preocupantes sobre o tempo de execução do algoritmo, destacando-se como pior caso a árvore de busca binaria.

Isto se deve ao fato de nossa árvore de busca binaria estão totalmente desbalanceada, de tal forma a complexidade do algoritmo se torna $O(n)$. Enquanto os dois métodos se enquadram na complexidade $O(\log n)$.

g. Houve diferença significativa entre o tempo de execução entre os três métodos avaliados no caso da ordenação dos vetores aleatórios? Se sim, qual foi o melhor método? Qual foi o pior? Justifique suas respostas.

Neste caso ordenamos 10 vetores em cada método e recolhemos sua média, desta forma, podemos perceber que gerando números aleatórios, nossa árvore não fica tão desbalanceada. Tal característica nos retorna um algoritmo de complexidade $O(\log n)$.

O algoritmo que levou mais tempo para ser executado foi o da árvore ARN, isso se deve ao fato de realizarmos rotações durante a inserção de novas chaves.

h. Em cada um dos métodos (usando ABB, AVL, ARN), o tempo de execução foi muito diferente entre a ordenação dos vetores inicialmente em ordem decrescente e

dos vetores aleatórios? Em qual método houve a maior variação? E a menor variação?

Podemos perceber uma diferença gritante na árvore ABB, pois como citamos na resposta acima, utilizando um vetor em ordem decrescente, a mesma nos retorna uma árvore totalmente desbalanceada. Por outro lado, ao utilizarmos chaves aleatórias, conseguimos montar uma árvore mais balanceada, resultando assim em um algoritmo mais rápido de ser computado.

A árvore que teve a menor variação foi a AVL, pois a mesma é uma árvore de balanceamento automático, resultando assim um algoritmo de complexidade $O(\log n)$ em todos os casos.

i. No item anterior você percebeu que em um dos métodos a variação do tempo de execução entre a ordenação de vetores inicialmente em ordem decrescente e dos vetores aleatórios é grande. Como você pode diminuir essa diferença?