

ITEM C

| | 1000 | 10000 | 100000 | 500000 |
|-----|----------|----------|-----------|----------|
| ABB | 0.004000 | 0.379000 | 77.363412 | time out |
| ARN | 0.000380 | 0.004890 | 0.076130 | 0.327483 |

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

| | 1000 | 10000 | 100000 | 500000 |
|-----|------|-------|--------|----------|
| ABB | 999 | 9999 | 99999 | time out |
| ARN | 9 | 13 | 16 | 18 |

Figure 2: Altura das Árvores Antes do Percorso Em-Ordem

ITEM D

| | 1000 | 10000 | 100000 | 500000 |
|-----|-------------------------|-------------------------|-------------------------|-------------------------|
| ABB | 0.00302 +- 0.000009 | 0.004425 +- 0.000034 | 0.092748 +- 0.003029 | time out |
| ARN | 0.000371 +- 0.000005 | 0.006577 +- 0.000297 | 0.128030 +- 0.004640 | 0.768600 +- 0.012932 |

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

| | 1000 | 10000 | 100000 | 500000 |
|-----|------|-------|--------|----------|
| ABB | 22 | 29 | 39 | time out |
| ARN | 8 | 11 | 14 | 15 |

Figure 4: Altura das Árvores Antes do Percorso Em-Ordem

ITEM E:

Sim, houve uma diferença bem significativa entre os resultados dos testes do item C. De acordo com o algoritmo, o melhor método utilizado foi o de Árvore Rubro Negra, uma vez que, devido a sua altura relativamente menor em relação a árvore de busca binária comum, seu tempo de execução também foi menor.

ITEM F

Houve uma diferença entre os métodos do item D, porém não tão significativa em relação aos itens anteriores. No entanto, percebe-se que, novamente, que o método ARN é muito mais rápido que o método de ABB.

ITEM G

Sim, houve uma grande diferença entre os métodos ABB e ARN em relação aos diferentes testes (vetores ordenados e vetores aleatórios). No entanto, o método que obteve uma diferença muito maior e, consequentemente uma significativa variação em relação ao outro foi o método ABB, sendo que o método ARN teve uma variação menor pois seu algoritmo, por natureza, é mais rápido que a ABB, portanto seus valores se mantiveram num intervalo reduzido tanto trabalhando com vetores ordenados quanto aleatórios.

ITEM H

O tempo de execução pode ser diminuído através da diminuição das iterações/chamadas de função. Isso pode ser feito criando um algoritmo que sempre desordene os valores de um vetor antes de adicionar os valores à árvore de busca binária.