Universidade Estadual de Santa Cruz
Conceitos e Linguagens de Programação
Larissa de Brito Santos - 202210349

1. Ordenação Imperativa

Código em C

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>


#define SIZE 100
#define NUM_VECTORS 1024
#define REPET 1024


void insertSort(int vector[], int n) {
for (int i = 1; i < n; i++) {
int chave = vector[i];
int j = i - 1;


while (j >= 0 && vector[j] > chave) {
vector[j + 1] = vector[j--];
}
vector[j + 1] = chave;
}
}


void randomVectorInt(int vector[], int n) {
for (int i = 0; i < n; i++) {
vector[i] = rand() % 1000; // Números aleatórios entre 0 e 999
}
}


int main() {
int qtd = REPET;
clock_t start, end;
double cpu_time_used;
```

```
FILE *arquivo = fopen("temposEmC.txt", "w");


while(qtd--){
int numVectors = NUM_VECTORS;
start = clock();
while(numVectors--){
int vector[SIZE];
randomVectorInt(vector, SIZE);
insertSort(vector, SIZE);
}
end = clock();
cpu_time_used = ((double) (end - start)) / CLOCKS_PER_SEC;


fprintf(arquivo, "%f\n", cpu_time_used);
}
fclose(arquivo);


return 0;
}
```

Linha de comando para compilação em C

```
gcc atv1a.c -o atv1a
```

2. Ordenação OO

Código em Java

```
import java.io.FileWriter;
import java.io.IOException;
import java.util.Random;


public class Atv1b {
private static final int SIZE = 100;
private static final int NUM_VECTORS = 1024;
private static final int REPET = 1024;
```

Universidade Estadual de Santa Cruz
Conceitos e Linguagens de Programação
Larissa de Brito Santos - 202210349

```
public static void insertSort(int[] vector, int n) {
for (int i = 1; i < n; i++) {
int chave = vector[i];
int j = i - 1;


while (j >= 0 && vector[j] > chave) {
vector[j + 1] = vector[j--];
}
vector[j + 1] = chave;
}
}


public static void randomVectorInt(int[] vector, int n) {
Random rand = new Random();
for (int i = 0; i < n; i++) {
vector[i] = rand.nextInt(1000); // Números aleatórios entre 0 e 999
}
}


public static void main(String[] args) {
int qtd = REPET;
long start, end;
double cpu_time_used;


try {
FileWriter arquivo = new FileWriter("temposEmJava.txt");


while (qtd-- != 0) {
int numVectors = NUM_VECTORS;
start = System.nanoTime();
```

```
while (numVectors-- != 0) {
int[] vector = new int[SIZE];
randomVectorInt(vector, SIZE);
insertSort(vector, SIZE);
}


end = System.nanoTime();
cpu_time_used = (double) (end - start) / 1_000_000_000; // Convertendo para segundos


arquivo.write(cpu_time_used + "\n");
}


arquivo.close();


} catch (IOException e) {
e.printStackTrace();
}
}
}
```

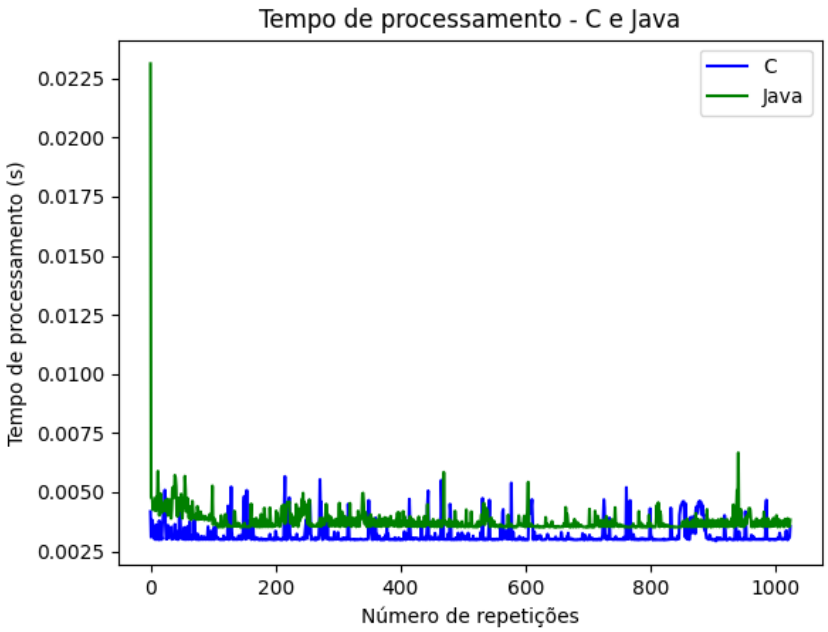
Linha de comando para compilação em Java

```
javac Atv1b.java
```

Universidade Estadual de Santa Cruz
Conceitos e Linguagens de Programação
Larissa de Brito Santos - 202210349

3. Comparação

No primeiro teste de cada linguagem os resultados saíam com bastantes picos de tempo, por isso refiz os testes algumas vezes antes de chegar com o resultado final.



Como apresentado acima, nos meus testes, o C se saiu mais rápido do que o Java na atividade proposta. Segue nas próximas páginas as tabelas: