



Trabalho Prático - Parte I | Relatório

Aluno: Lucas Rodrigues Oliveira
Matrícula: 475053
Curso: Ciência da Computação

Nesse trabalho foi implementado na linguagem C dois algoritmos de ordenação, chamados de **Insertion Sort** e **Heap Sort**. Ambos foram criados em um único arquivo, denominado por `source.c`. A ideia gira em torno do vetor `vetor_original[]` que armazena um tamanho N em que seus valores são gerados aleatoriamente. Além disso, durante a execução este vetor copia seus dados para outro vetor `vetor_novo[]` para que seja possível usar o mesmo tamanho e números tanto para o Insertion Sort quanto para o Heap Sort. Após a ordenação o programa mostra o tempo obtido por cada algoritmo, que foi calculado por meio da função `clock()`, presente na biblioteca `<time.h>`.

Insertion Sort

O Insertion Sort se trata de um algoritmo de ordenação simples que funciona de maneira semelhante à maneira como você classifica as cartas de baralho em suas mãos. A matriz é virtualmente dividida em uma parte classificada e uma parte não classificada. Os valores da peça não classificada são selecionados e colocados na posição correta na peça classificada. Vale ressaltar que a ordenação por inserção é eficiente para pequenos valores de dados, visto que quando se trata de valores maiores ou quando são classificados em ordem inversa, o algoritmo leva mais tempo para classificá-los.

Complexidade de tempo: $O(N^2)$

Heap Sort

O Heap Sort é uma técnica de ordenação baseada em comparação com base na Binary Heap. Esta técnica é semelhante ao Select Sort, onde primeiro encontramos o elemento mínimo e colocamos o elemento mínimo no início. Repita o mesmo processo para os demais elementos.

A diferença dele para o Insertion Sort é que o tempo necessário para realizar a ordenação aumenta logaritmicamente, enquanto o Insertion cresce exponencialmente mais lento. Além disso o uso de memória é mínimo, visto que, além do necessário para manter a lista inicial de itens a serem classificados, não precisa de espaço de memória adicional para funcionar.

Complexidade de tempo: $O(N \log N)$

Tempo de Execução

Tamanho do vetor	Heap Sort	Insertion Sort
1.000	2 ms	5 ms
10.000	10 ms	79 ms
100.000	44 ms	5668 ms
1.000.000	537 ms	597813 ms



Para valores acima de 1 M, o programa apresenta a seguinte mensagem:

```
main.c:80:14: warning: passing argument 1 of 'free' makes pointer from integer without a cast [-Wint-conversion]
  80 |         free(elem);
      |         ^~~~~~
      |         |
      |         int
In file included from main.c:2:
/usr/include/stdlib.h:555:25: note: expected 'void *' but argument is of type 'int'
  555 | extern void free (void *__ptr) __THROW;
      |                   ^~~~~~
```

Gráfico de Execução

