

Universidade de Brasília  
Instituto de Ciências Exatas  
Departamento de Ciência da Computação

117536 - Projeto e Análise de Algoritmos  
Turma: B

Relatório sobre **Formalização de  
Propriedades do Algoritmo Radix  
Sort utilizando Prototype Verification  
System (PVS)**

Davi Assis Martinez - 17/0140113  
João Antonio D. de Moraes - 16/01269575  
Matheus Lemos - 16/0137969

13 de novembro de 2019

## **1 Introdução**

O presente relatório é relativo ao projeto do segundo semestre da disciplina de Lógica Computacional 1, onde se procurou demonstrar a corretude do algoritmo de ordenação *radixsort*.

## 2 Apresentação do Problema

O estudo dos algoritmos de ordenação é essencial à Ciência da Computação. Seja em contexto acadêmico ou apenas em atividades mundanas, lidamos constantemente com tarefas e informações que exigem organização a partir de um critério - alfabético, numérico crescente ou decrescente, cronológico, entre outros.

Existem diversos métodos para obter a ordenação de uma lista de objetos. Entre eles, há algoritmos que implementam estabilidade dos dados ordenados, ou seja, elementos iguais mantêm a mesma posição relativa entre eles que possuíam antes da ordem. É importante considerar a necessidade de um algoritmo estável: se uma lista será ordenada várias vezes até atingir a ordenação desejada, não queremos que cada iteração desorganize o trabalho da iteração anterior.

Na questão 2 foi realizada uma prova expandindo-se a definição de *radixsort*, o que implica em utilizar as definições do *merge\_sort*, utilizando a ordem "estritamente menor". Mais especificamente, foi desenvolvida uma prova sobre a o lema *merge\_sort\_is\_permutation*, que diz que o *merge\_sort* de uma lista é uma permutação dos mesmos elementos da lista. Ou seja, o conteúdo da lista permanece o mesmo. Após expandidas essas definições foi aplicada a definição de *permutations*, estabelecida no arquivo *mergesort.pvs*. Essa definição introduz que para todo  $x:T$  pertencente a lista  $l$ , qualquer ocorrência no *merge\_sort*( $l$ ) também está em  $l$ . Após feitas as devidas instanciações, foi introduzido novamente o lema *merge\_sort\_is\_permutation*, dessa vez com a ordem "menor ou igual". Essa instanciação foi feita de modo a substituir o sequente "*merge\_sort*( $l$ )" por "*merge\_sort*(*merge\_sort*[ $T, i=$ ]( $l$ ))", que é o sequente que se deseja encontrar, o que finaliza a prova.

### 2.1 Análise assintótica

### 2.2 Correção da solução proposta

## 3 Conclusão