

# Lista I

## Programação Concorrente e Distribuída

Rodrigo Caetano Rocha

26 de agosto de 2015

**Valor:** 5 pontos

**Entrga:** 02/Set.

### 1 O problema da contagem de primos

**valor:** 2 pt.

**Descrição**

Um número primo é um inteiro positivo maior que um, sendo divisível apenas por um e o próprio número. Existem infinitos números primos. Apesar de ser um conceito abstrato de teoria dos números em matemática, números primos começaram a ganhar grande importância fora do contexto de teoria dos números, passando a ter grandes aplicações, sendo a de maior destaque em criptografias modernas, na área de segurança da informação.

O problema da contagem de números primos<sup>1</sup> consiste em calcular quantos números primos existem até um determinado valor limite  $n$ . Essa função é denotada por  $\Pi(n)$ , para  $n \geq 1$ . A figura 1 ilustra a função  $\Pi(n)$ .

**Questões**

1. Apresente uma solução paralela sem utilizar o reduction.
2. Apresente uma solução paralela utilizando o reduction.
3. Analise o *speedup* e a eficiência das soluções paralela em relação ao sequencial (informe o número de núcleos e qual foi o processador utilizado na análise).
4. Explique porque a contagem de primos possui desbalanceamento de carga.
5. Descreva uma possível melhoria na distribuição da carga entre as threads.

---

<sup>1</sup>Em matemática, o problema da contagem de números primos está fortemente ligado à Hipótese de Riemann, um dos Problemas do Prêmio do Milênio oferecido pelo Instituto Clay de Matemática, cujo valor do prêmio é de um milhão de dólares.

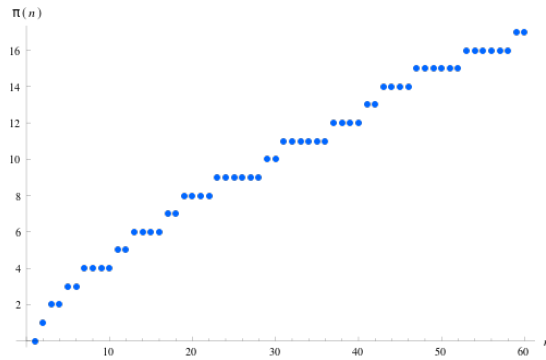


Figura 1: Visualização da função de contagem de primos,  $\Pi(n)$ , para alguns valores iniciais de  $n$ .

## 2 Ordenação com o MergeSort

**valor:** 3 pts.

**Descrição**

O merge sort é um algoritmo de ordenação baseado no modelo de divisão e conquista. A primeira etapa consiste em dividir a entrada em porções unitárias. Uma vez que a entrada foi dividida, as porções são combinadas aos pares, produzindo uma porção maior ordenada. A figura 2 apresenta um exemplo da execução do merge sort.

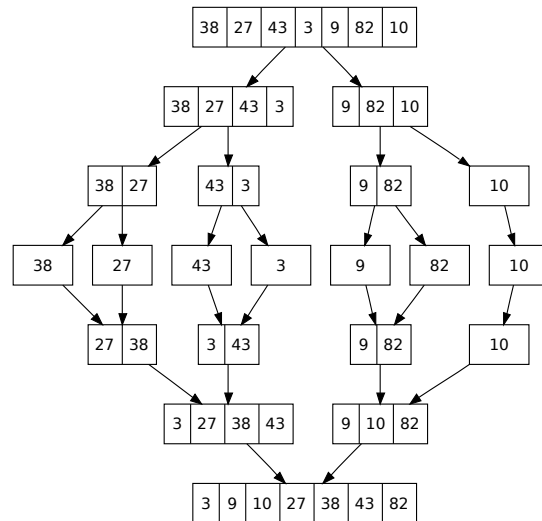


Figura 2: Árvore de recursão para a execução em um exemplo do merge sort.

Por ser um algoritmo conceitualmente recursivo, existem alguns desafios ao paralelizar o merge sort. Entretanto, a computação realizada em cada ramo da árvore de recursão é independente aos demais ramos, e portanto paralelizável.