Construção e Análise de Algoritmos

Lista de exercícios 02

1. Apresente uma implementação não recursiva do algoritmo Mergesort.

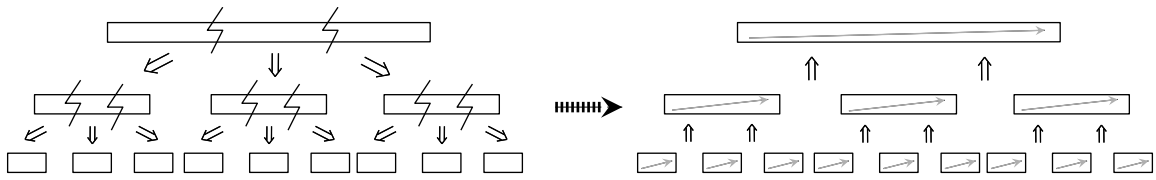
   Dicas:

   - utilize uma variável `bloco` que indica o tamanho dos blocos que estão sendo intercalados em cada etapa, cujo valor começa em 1 e é dobrado sucessivamente até alcançar $n/2$;
   - utilize variáveis auxiliares `i1` e `i2` que indicam o início dos pares de intervalos que serão intercalados a cada passo

   Estime o tempo de execução do seu algoritmo.

2. **3-Mergesort**

   Imagine que alguém decide modificar o algoritmo Mergesort para que ele divida a lista em 3 partes iguais a cada passo.



   a) Apresente o pseudocódigo do procedimento `3-Intercalação` e do procedimento principal do algoritmo `3-Mergesort`.

   b) Estime o tempo de execução do algoritmo `3-Mergesort`.

   c) Você acha que, na prática, essa variante executa mais rápido que o algoritmo original? Porque?

3. **Mergesort paralelo** (OPCIONAL)

   Uma das vantagens da técnica de divisão é que, ao quebrar o problema original em múltiplos subproblemas, esses subproblemas podem ser resolvidos em paralelo por uma máquina com múltiplos processadores.

   Imagine que o algoritmo Mergesort é adaptado para executar em uma máquina com múltiplos processadores, para aproveitar ao máximo as oportunidades de paralelização.

   Mais especificamente, note que todo trabalho de ordenação da lista é realizado pelo procedimento `Intercalação`. Então, a ideia é que, sempre que possível, as chamadas a esse procedimento são executadas em paralelo.

   a) Estime o tempo de execução do algoritmo em uma máquina com 2 processadores.

   b) Estime o tempo de execução do algoritmo em uma máquina com n processadores.

4. **Algoritmo de desordenação** (OPCIONAL)

   Suponha que a linha `3` do procedimento `Intercalação` é trocada pelo seguinte fragmento de código:

   ```
   3′.        lance uma moeda honesta
   3″.        Se ( o resultado é cara )  {
   ```

   Agora, imagine que o algoritmo Mergesort é executado com essa variante do procedimento `Intercalação`.

   Esse algoritmo produz uma configuração perfeitamente aleatória da lista de entrada?

   (i.e., todas as configurações possíveis da lista têm a mesma probabilidade de serem produzidas pelo algoritmo?)

   Dica: Examine o comportamento do algoritmo com listas de tamanho 2,3,4.