

Projeto e Análise de Algoritmos (2021-2)

Lista de exercícios (Relações de recorrência)

Prof. Flávio L. C. de Moura

19 de fevereiro de 2022

1. Considere a versão recursiva do algoritmo de ordenação por inserção que formalizamos no Coq.

```
Fixpoint ord_insercao l :=
  match l with
  | nil => nil
  | h :: tl => insere h (ord_insercao tl)
  end.
```

onde

```
Fixpoint insere x l :=
  match l with
  | nil => x :: nil
  | h :: tl => if (x <=? h) then x :: l else h :: (insere x tl)
  end.
```

- (a) Escreva a relação de recorrência que corresponde ao custo de execução da função `insere`, *i.e.* que computa o número $T_{ins}(n)$ de comparações feitas para inserir um elemento em uma lista com n elementos.
- (b) Resolva a relação de recorrência do item anterior.
- (c) Escreva a relação de recorrência que corresponde ao custo de execução da função `ord_insercao`, *i.e.* que computa o número $T(n)$ de comparações feitas para ordenar uma lista com n elementos.
- (d) Resolva a relação de recorrência do item anterior.
2. Resolva as seguintes relações de recorrência:
- (a) $T(1) = 1, T(n) = 3T(n/2) + n^2, n \geq 2$
- (b) $T(1) = 1, T(n) = 2T(n-1) + 1, n \geq 2$
- (c) $T(1) = 1, T(n) = 2T(n/2) + n, n \geq 2$
- (d) $T(1) \in \Theta(1), T(n) = 3T(n/2) + n \ln(n)$
- (e) $T(1) \in \Theta(1), T(n) = 3T(n/3 + 5) + n/2$
- (f) $T(1) \in \Theta(1), T(n) = 2T(n/2) + n/\ln(n)$
- (g) $T(1) \in \Theta(1), T(n) = T(n-1) + 1/n$
- (h) $T(1) \in \Theta(1), T(n) = T(n-1) + \ln(n)$
- (i) $T(1) \in \Theta(1), T(n) = \sqrt{n}T(\sqrt{n}) + n$
3. A relação de recorrência descrita no enunciado do teorema mestre caracteriza a estratégia de “divisão e conquista” utilizada na construção de alguns algoritmos. Esta estratégia consiste em dividir o problema original em subproblemas menores, resolver cada um destes subproblemas e combinar estas soluções em uma solução do problema original. A função de exponenciação é definida por:

$$a^n = \begin{cases} 1, & \text{se } n = 0; \\ a \cdot a^{n-1}, & \text{se } n > 0 \end{cases}$$

- Implemente um algoritmo que utilize a estratégia de dividir e conquistar para calcular a^n , onde n é inteiro positivo;
 - Construa e resolva a relação de recorrência que computa do número de multiplicações executadas por este algoritmo;
 - Como este algoritmo se compara ao algoritmo força-bruta (implementação direta da definição)?
4. O problema das torres de Hanoi consiste em transladar n discos de diâmetros diferentes de uma torre A a uma torre C usando uma torre auxiliar B . Os discos estão inicialmente organizados na torre, A , decrescentemente, segundo o diâmetro e devem terminar na torre C com a mesma organização. Os discos são movidos de uma torre a outra com a restrição de que jamais se têm discos de diâmetro maior acima de discos de diâmetro menor numa mesma torre. Ademais, em um movimento não podem ser transladados varios discos simultaneamente e somente é permitido mover discos no tope das torres.

A solução típica do problema (veja Figura 1), consiste em:

- Transladar recursivamente os $n - 1$ discos de menor diâmetro, da torre A à torre B , usando a torre C como torre auxiliar;
- Mover o disco de maior diâmetro de A a C ;
- Transladar recursivamente os $n - 1$ discos de B a C , usando a torre A como torre auxiliar.

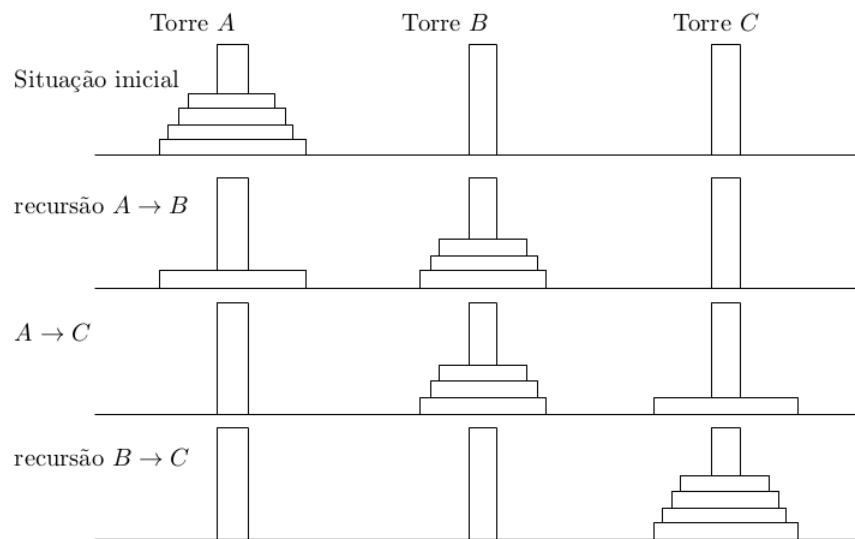


Figura 1: Torres de Hanoi

Seja $M(n)$ o número de movimentos necessários para transladar n discos segundo o algoritmo anterior.

- Construa a relação de recorrência abaixo expressa o número correto de movimentos $M(n)$;
- Resolva a relação de recorrência do item anterior.