Concluída

- Considere árvores binárias de pesquisa, em OCaml.

  1. Considere o tipo ``'a abp`` que representa uma árvore binária de pesquisa com elementos do tipo 'a; defina esse tipo.
  2. Defina uma função ``lookup`` com o tipo ``'a abp -> 'a -> bool`` que devolve true se o segundo argumento estiver na árvore que é o primeiro argumento.
  3. Defina uma função ``insert`` com tipo ``'a abp -> 'a -> 'a abp`` que insere o segundo argumento na árvore que é o primeiro argumento, dando uma nova árvore (o resultado)
  4. Defina uma função ``delete`` com o mesmo tipo que a anterior.
  5. Defina ua função ``walk`` com tipo ``'a abp -> ('a -> unit) -> unit`` que atravessa a árvore do primeiro argumento em ordem e chama função que é o segundo argumento sobre cada valor da árvore.

Última alteração: terça, 13 de dezembro de 2022 às 11:54