**Questão 1** : Uma árvore é uma estrutura tipicamente recursiva. Nós vimos um algoritmo iterativo para realizar busca. Porque não utilizar um algoritmo recursivo para a busca em uma árvore binária de busca?

**Resposta** : A resposta pra isso não está na complexidade, mas no uso da memória. A complexidade de realizar a busca, tanto com iteração quanto com recursão, no pior caso será a mesma, pois o que garante a melhor complexidade na busca em árvores é justamente pela estrutura ser uma árvore, então em qualquer busca, ela sempre deixará de ver pelo menos metade dos nós a cada vez que iterar, o que irá fazer com que sejam lidos nós de forma logaritmica. Entretanto, na iteração, apenas realizamos a operação própriamente dita e partimos para o próximo nó a ser lido, enquanto na recursão, nós iremos empilhar na stack do sistema a própria função e partir para o próximo nó, até encontrar o nó correto e depois disso, ir desempilhando as funções até voltar para a raíz. Dessa forma, a recursão irá utilizar muito mais memória do que a iteração na busca de um elemento na árvore e pode causar problema de falta de memória, pois é gasto quase o dobro de memória da iteração, pois temos o mesmo número de iterações que o método iterativo e mais o desempilhamento da função até retornar à raíz, o que pode causar esse problema de falta de memória dependendo do tamanho da estrutura e do elemento que estamos buscando. E um último problema que pode ser gerado é também o problema de performance, pois serão geradas mais ações no método recursivo do que no iterativo.