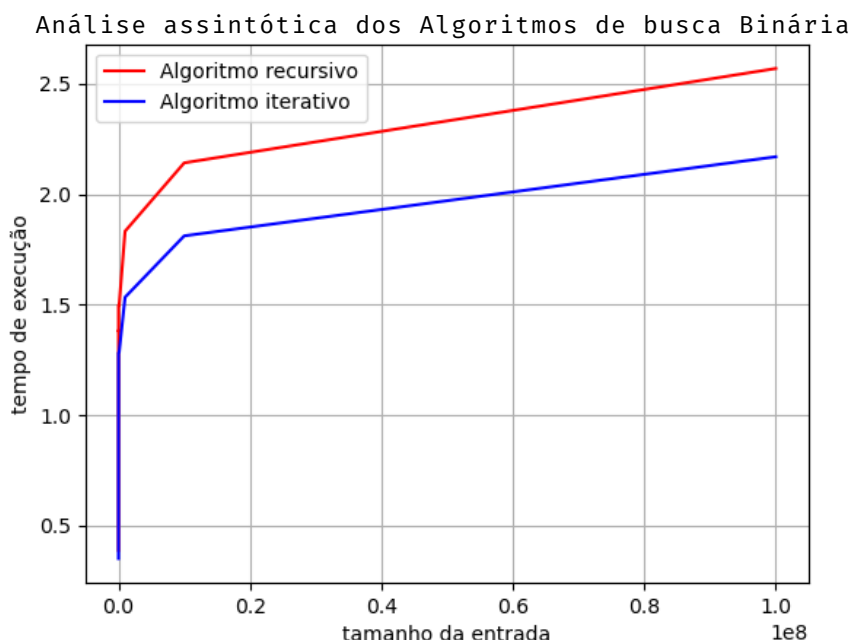


Escreva um algoritmo (em pseudocódigo) que realize busca binária de forma iterativa e o implemente numa linguagem de programação a sua escolha. Construa um gráfico mostrando a relação valor de entrada x tempo de execução do algoritmo implementado. Considerando uma análise assintótica em pior caso, explique se o desempenho do algoritmo implementado é superior, inferior ou igual ao do algoritmo que implementa busca binária de forma recursiva.

```
BINARY-SEARCH(listaOrdenada, valorProcurado, menor, maior):
01.     indiceDoElementoDoMeio = (menor + maior) // 2
02.     elementoDoMeio = listaOrdenada[indiceDoElementoDoMeio]
03.
04.     encontrado = False
05.     Enquanto não encontrado faça:
06.         Se (menor = maior) então:
07.             Se (elementoDoMeio ≠ valorProcurado) então:
08.                 Retorne Nulo
09.             Senão
10.                 encontrado = True
11.         Senão Se (elementoDoMeio = valorProcurado) então:
12.             encontrado = True
13.         Senão Se (elementoDoMeio > valorProcurado) então:
14.             novaPosicao = indiceDoElementoDoMeio - 1
15.             maior = novaPosicao
16.             indiceDoElementoDoMeio = (menor + maior) // 2
17.             elementoDoMeio = listaOrdenada[indiceDoElementoDoMeio]
18.             Se (elementoDoMeio = valorProcurado) então:
19.                 encontrado = True
20.         Senão Se (elementoDoMeio < valorProcurado) então:
21.             novaPosicao = indiceDoElementoDoMeio + 1
22.             menor = novaPosicao
23.             indiceDoElementoDoMeio = (menor + maior) // 2
24.             elementoDoMeio = listaOrdenada[indiceDoElementoDoMeio]
25.             Se (elementoDoMeio = valorProcurado) então:
26.                 encontrado = True
27.     Retorne indiceDoElementoDoMeio
```



Conforme a análise empírica dos dois algoritmos, percebe-se que ambos apresentam complexidade logarítmica, ou seja, assintoticamente eles são equivalentes. No entanto, aparentemente, o algoritmo iterativo apresenta um desempenho um pouco melhor (na prática) por utilizar loops simples, já o algoritmo recursivo, por fazer múltiplas chamadas de função, ele precisa alocar espaço para as variáveis locais e empilhar argumentos, ou seja, por utilizar uma estrutura de dados para empilhar isso, a execução acaba tendo uma diferença na prática, causando um pequeno overhead comparado à execução de um loop simples.