Lenguajes de Programación

Práctica Calificada de Haskell - Funciones

1. En el siguiente artículo, <https://www.researchgate.net/publication/288832602_Ternary_Search_Algorithm_Improvement_of_Binary_Search>

Se probó que el algoritmo de búsqueda ternario supera al algoritmo de búsqueda binaria. Pruebe si el autor estaba realmente en lo correcto o no.

* Construya el algoritmo de búsqueda ternario pero utilizando recursión.
* Utilice el algoritmo de búsqueda binario realizado en la anterior clase.
* Compare el tiempo de ambos algoritmos.
* Para las pruebas:
  + Pruebe hacer pruebas con un arreglo [1,4.. N] donde N pueden ser valores 1mil, 5 mil, 10mil … 100 mil elementos.
  + Pruebe con la búsqueda de valores aleatorios entre 0 y 2\*N, una cantidad N/2 de veces.
  + Para cada prueba, marque el tiempo de ejecución
  + Construya una tabla de la forma:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| N | 1 mil | 5 mil | 10 mil | 50 mil | 100 mil |
| Tiempo Binario | 0.02 | 0.35 | 1.19 | 29.56 | 124.42 |
| Tiempo  Ternario | 0.02 | 0.30 | 0.97 | 28.13 | 102.47 |

* + Grafique ambas funciones utilizando cualquier herramienta y presente la comparación.

