

Estrutura de Dados e Algoritmos 2: Lista 01

Maurício Serrano

Paulo Tada 11/0135431 Maria Luciene 12/0037742

Exercício 1

O código abaixo é uma versão modificada de um algoritmo de busca binária utilizando vetor. A alteração insere o método da interpolação (linha 13).

Listing 1: C++ Code

```
1  // Binary Search
2  #include <iostream>
3  #include <stdlib.h>
4  #include <ctime>
5
6  using namespace std;
7
8  // Binary search adding interpol method
9  int binary_search(int vector[], int element, int min, int max){
10     int result = -1;
11     while(min<=max){
12         // Modification to use interpol on binary search
13         int mid = min + (max - min)*((element-vector[min])/(vector[max]-vector[min]));
14         if(vector[mid] == element){
15             result = mid;
16             break;
17         } else
18         if(vector[mid] < element){
19             min = mid + 1;
20         } else {
21             max = mid - 1;
22         }
23     }
24     return result;
25 }
26
27 void preconfig(int vector[], int max_size){
28     for(int i = 0 ; i<max_size ; i++){
29         vector[i] = i*2;
30     }
31 }
32
33 int main(){
34     int max_size = 0; // Max size of the vector
35     while(1){
36         cin >> max_size;
37         if(max_size==-1) break;
38
39         int vector[max_size];
40         preconfig(vector, max_size); // Creates a vector with multiples of 2
41         int element = 0; // Element to find
42         cin >> element;
43
44         clock_t start, end;
45         start = clock();
46         binary_search(vector,element,0,max_size-1);
47         end = clock();
```

```
48
49     long double tmili = (end - start)/((double) (CLOCKS_PER_SEC/1000));
50     cout << "Time: " << tmili << " ms" << endl;
51
52 }
53 return 0;
54 }
```

Com a execução do código com os casos de teste com vetores de tamanho: 10, 25, 50, 100, 500, 1000, 10000, 100000.

Listing 2: Input files for test cases

```
1 10
2 12
3 25
4 12
5 50
6 12
7 100
8 12
9 500
10 12
11 1000
12 12
13 10000
14 12
15 100000
16 12
17 1000000
18 12
19 -1
```

Após a execução do programa temos o tempo da busca em cada vetor.

Listing 3: Output file for test cases

```
1 Time: 0.004 ms
2 Time: 0.003 ms
3 Time: 0.003 ms
4 Time: 0.003 ms
5 Time: 0.002 ms
6 Time: 0.003 ms
7 Time: 0.003 ms
8 Time: 0.001 ms
9 Time: 0.002 ms
```

Exercício 2