

# Relatório: Bogosort em C

---

## Introdução

Este relatório apresenta uma análise do algoritmo Bogosort implementado em C. Bogosort é um algoritmo de ordenação ineficiente que embaralha os elementos de um array repetidamente até que eles estejam ordenados. Embora tenha um valor educacional para ilustrar algoritmos de ordenação, seu tempo de execução é extremamente elevado em casos práticos, sendo considerado uma abordagem impraticável. Aqui, realizamos um teste de mesa utilizando o seguinte array de entrada: {7, 8, 1, 6, 14, 9, 2, 10, 3, 4, 13, 5, 0, 11, 12}.

## Algoritmo Bogosort em C

O algoritmo Bogosort é simples: ele embaralha o array até que os elementos estejam na ordem correta. Para verificar se o array está ordenado, ele percorre o array e compara cada elemento com o próximo. Se o array não estiver ordenado, o algoritmo o embaralha e repete o processo. Abaixo está uma explicação do código.

## Código em C:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

// Função para verificar se o array está ordenado
int isSorted(int *a, int n) {
    while (--n > 0) {
        if (a[n] < a[n-1]) return 0;
    }
    return 1;
}

// Função para embaralhar o array
void shuffle(int *a, int n) {
    for (int i = 0; i < n; i++) {
        int r = rand() % n;
        int temp = a[i];
        a[i] = a[r];
        a[r] = temp;
    }
}
```

```

// Função Bogosort
void bogosort(int *a, int n) {
    while (!isSorted(a, n)) {
        shuffle(a, n);
    }
}

// Função principal para testar o Bogosort
int main() {
    int a[] = {7, 8, 1, 6, 14, 9, 2, 10, 3, 4, 13, 5, 0, 11, 12};
    int n = sizeof(a)/sizeof(a[0]);

    srand(time(0));

    bogosort(a, n);

    for (int i = 0; i < n; i++) {
        printf("%d ", a[i]);
    }
    printf("\n");

    return 0;
}

```

## Teste de Mesa

A seguir, realizamos um teste de mesa com o array de entrada {7, 8, 1, 6, 14, 9, 2, 10, 3, 4, 13, 5, 0, 11, 12}. Como o algoritmo Bogosort depende de embaralhamentos aleatórios, o número de iterações pode variar. Para este teste, consideramos um cenário hipotético onde o array foi embaralhado diversas vezes até atingir a ordenação correta. As iterações mostradas abaixo são exemplos simplificados para demonstrar o comportamento do algoritmo.

### Iterações:

1ª Iteração: {7, 8, 1, 6, 14, 9, 2, 10, 3, 4, 13, 5, 0, 11, 12} (não ordenado)

2ª Iteração: {9, 7, 0, 5, 3, 1, 6, 2, 8, 14, 10, 12, 13, 11, 4} (não ordenado)

3ª Iteração: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14} (ordenado)

## **Conclusão**

O algoritmo Bogosort é extremamente ineficiente, principalmente para arrays maiores, como o exemplo dado. Devido à natureza aleatória do embaralhamento, ele pode demorar muito para ordenar uma lista, tornando-o impraticável para a maioria dos casos. Neste teste, o array foi embaralhado até que os elementos estivessem em ordem crescente. No entanto, o número de iterações necessárias pode variar drasticamente dependendo do sorteio dos embaralhamentos.