Basile Relatório Max-Heapfy

Mateus Felipe da Silveira Vieira

October 20, 2025

Repositório Git: https://github.com/habdig7oficial/max_heapfy.git

Código Heapsort Completo

```c
#include "stdio.h"
#include "math.h"
#include "stdbool.h"

void print_vec(int *vec, int len){
    printf("[");
    for (int i = 0; i < len; i++){
        printf("%d, ", vec[i]);
    }
    printf("]\n");
}

void swap(int *a, int *b){
    int aux = *a;
    *a = *b;
    *b = aux;
}

bool max_heapfy(int *vec, int i, int len){
    bool parity = (i % 2) == 0;
    int index_father = (int)i / 2 + (parity? -1 : 0);
    int index_brother = i + (parity ? -1 : 1);
    printf("Root: vec[%d] = %d\n", index_father, vec[index_father]);
    printf("Leaf: vec[%d] = %d\n", i, vec[i]);
    printf("Brother: vec[%d] = %d\n", index_brother, vec[index_brother]);
    printf("\n");

    if(index_brother >= len){
        return false;
    }
    int max_child = (vec[i] >= vec[index_brother])? i : index_brother;
    if(vec[max_child] > vec[index_father]){
        swap(&vec[index_father], &vec[max_child]);

        printf("Troca: %d %d\n", vec[max_child], vec[index_father]);
        return true;
    }

    print_vec(vec, len);
    return false;
}

int main(){
    int vec[] = {2, 14, 6, 8, 5, 4, 3, 1, 7, 9, 6, 10, 17, 20, 12, 19};
    int len = sizeof(vec) / sizeof(int);
    print_vec(vec, len);

    for(int i = len - 1; i > 0; i -= 2){
        max_heapfy(vec, i, len);
    }

    printf("\n----------End max Heapfy----------\n");

    print_vec(vec, len);

    for(int i = len - 1; i > 0; i--){
        swap(&vec[0], &vec[i]);
        for(int j = 1; max_heapfy(vec, j, i); j += 2){
            print_vec(vec, len);
        }
    }


    print_vec(vec, len);
}
```

Print de Execução