16 MIN and MaX using Divide And conquere

#include <stdio.h>

struct Pair {

int min;

int max;

};

struct Pair find\_max\_min(int data[], int low, int high) {

struct Pair result;

// Base case: Single element list

if (low == high) {

result.min = data[low];

result.max = data[low];

return result;

}

// Mid index calculation

int mid = low + (high - low) / 2;

// Recursive calls to find min and max in both halves

struct Pair left = find\_max\_min(data, low, mid);

struct Pair right = find\_max\_min(data, mid + 1, high);

// Combine results: minimum and maximum of both halves

result.min = (left.min < right.min) ? left.min : right.min;

result.max = (left.max > right.max) ? left.max : right.max;

return result;

}

int main() {

int data[] = {6, 5, 3, 1, 8, 7, 2, 4};

int n = sizeof(data) / sizeof(data[0]);

struct Pair result = find\_max\_min(data, 0, n - 1);

printf("Maximum value: %d\n", result.max);

printf("Minimum value: %d\n", result.min);

return 0;

} 