

# Atividade 7



Lucas de Araújo - 18.2.4049

```
/*
* Lucas de Araújo - 18.2.4049
*/
#include <stdio.h>

/*
- Análise Assintótica: getmax

*  $T(1) = 1$ 
*  $T(n) = 2 T(n/2) + 5 = 2 T(n/2) + O(1)$ 

- Utilizando teorema mestre:

*  $a = 2$ ,  $b = 2$  e  $d = 0$ 
*  $\log_2(2) > 0 \rightarrow O(n)$ 
*/

int getmax(int arr[], int low, int high)
{
    if (low == high) // 1
        return arr[low]; // 1

    int mid = (low + high) / 2; // 1

    int max1 = getmax(arr, low, mid); //  $T(n/2)$ 
    int max2 = getmax(arr, mid + 1, high); //  $T(n/2)$ 

    return max1 > max2 ? max1 : max2; // 2
}

/*
- Análise Assintótica: get_min_max

*  $T(1) = 1$ 
*  $T(n) = 2 T(n/2) + 9 = 2 T(n/2) + O(1)$ 

- Utilizando teorema mestre:
```

```

* a = 2, b = 2 e d = 0

* Log2(2) > 0 -> O(n)

*/

void get_min_max(int arr[], int low, int high, int *min, int *max)
{
    if (low == high) // 1
    {
        *min = *max = arr[low]; // 2
    }
    else
    {
        int mid = (low + high) / 2; // 1
        int min1, max1, min2, max2; // 1

        get_min_max(arr, low, mid, &min1, &max1); // T(n/2)
        get_min_max(arr, mid + 1, high, &min2, &max2); // T(n/2)

        *min = min1 < min2 ? min1 : min2; // 2
        *max = max1 > max2 ? max1 : max2; // 2
    }
}

/*

- Análise Assintótica: power

* T(0) = 1
* T(n) = 1 T(n/2) + 8 = 1 T(n/2) + O(1)

- Utilizando teorema mestre:

* a = 1, b = 2 e d = 0

* Log2(1) > 0 -> O(1 * log(n)) -> O(log(n))

*/

long long int power(int base, unsigned int exp)
{
    if (exp == 0) // 1
        return 1; // 1

    long long int temp = power(base, exp / 2); // T(n/2)

    if (exp % 2 == 0) // 2
        return temp * temp; // 2
    else
        return base * temp * temp; // 2
}

int main(int argc, char const *argv[])
{
    int arr[] = {1, 45, 23, 67, 2, 88, 17, 33};

```

```

int n = sizeof(arr) / sizeof(arr[0]);
printf("O maior valor no vetor é %d\n", getmax(arr, 0, n-1));

int min, max;
get_min_max(arr, 0, n-1, &min, &max);
printf("O menor valor no vetor é %d\n", min);
printf("O maior valor no vetor é %d\n", max);

int base = 2;
unsigned int exp = 10;

printf("%d Elevado a %d = %lld\n", base, exp, power(base, exp));

return 0;
}

```