- int — [1, 2, 3] → int x[]
- long int
- char — "mateus\0" → char s[]
- bool — ['m', 'A', 't', 'e', 'u', 's', '\0']

# Análise de Algoritmos:

- Verificar quanto de recurso será necessário pelo computador para resolver o problema

- Tempo
- Memória
- Notação Big O (Pior caso)

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 9 | 7 | 6 | 5 | 2 | 3 | 4 | 8 | 10 |

- find(x) → indice (Posição de x)
- melhor caso: find(1) → 0
- médio caso: find(2) → 5
- pior caso: find(10) → 9

- N → inteiro (1 até 10)
- vetor → N de tamanho

- O(f(x))
- Constante: O(1)

```
int main() {
    int x;
    scanf("%d", &x);
    printf("%d\n", x);
}
```

- linear: O(x)

```
int main() {
    int x, sum = 0;
    scanf("%d", &x);
    for(int i = 1; i <= x; i++) {
        sum += i;
    }
}
```

- Quadrática: $O(n^2)$

```
int main(){
    int n, sum = 0;
    scanf("%d", &n);
    for(int i = 1; i <= n; i++){
        for(int k = 1; k <= n; k++){
            sum += i;
        }
    }
}
```

- cúbica: $O(n^3)$
- Exponencial: $O(2^N)$

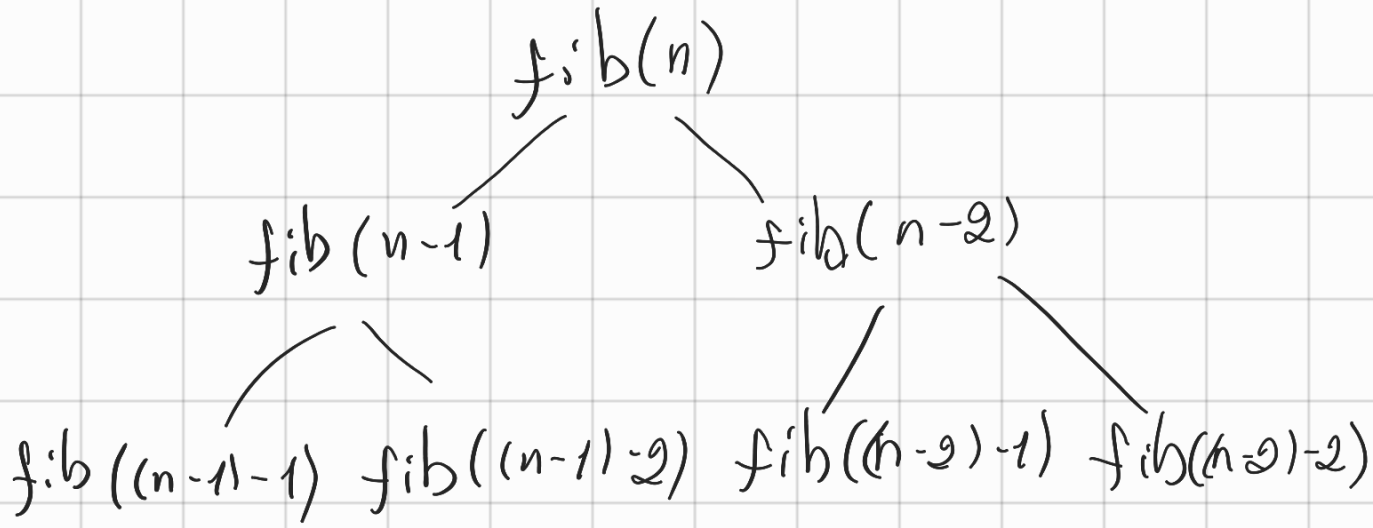
$N \rightarrow$ nº de VAR

| | A | b |
|---|---|---|
| 1 | T | F |
| 2 | T | T |
| 3 | F | F |
| 4 | F | T |

| | A | b | c |
|---|---|---|---|
| 1 | F | F | F |
| 2 | F | F | T |
| 3 | F | T | F |
| 4 | F | T | T |
| 5 | T | F | F |
| 6 | T | F | T |
| 7 | T | T | F |
| 8 | T | T | T |

```
int fib(int n){
    if(n <= 1) return n;
```

```
    return fib(n-1) + fib(n-2);
}
```




```
int f(int A) {
    if (A == 0) return 1;
    return A + f(A - 1);
}
```

A = 5

f(5) → f(4) → f(3) → f(2) → f(1)

0 1 2 3 4 5 6 7 8 9

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

- Log: O(log N)

```
int find(int x, int v[], int left, int right) {
    int mid = (right + left) / 2;
    if (v[mid] == x) return mid;
    if (left >= right) return -1;
```

```
    else if( V[mid] > x)
        return find(x, V, left, mid-1);
    else
        return find(x, V, mid+1, high);
}
```

- $O(\log N)$, sendo $N$ o número de elementos do vetor.

$$T(n) = T\left(\frac{n}{2}\right) + 1$$

$$= \left(T\left(\frac{n}{4}\right) + 1\right) + 1$$

$$= \left[\left(T\left(\frac{n}{8}\right) + 1\right) + 1\right] + 1$$

$$= \Theta(n^k \log n); \quad K = 0$$

$$= O(\log n)$$

[33, 51, 23, 94, 66, 28, 11, 73, 19, 8, 31]

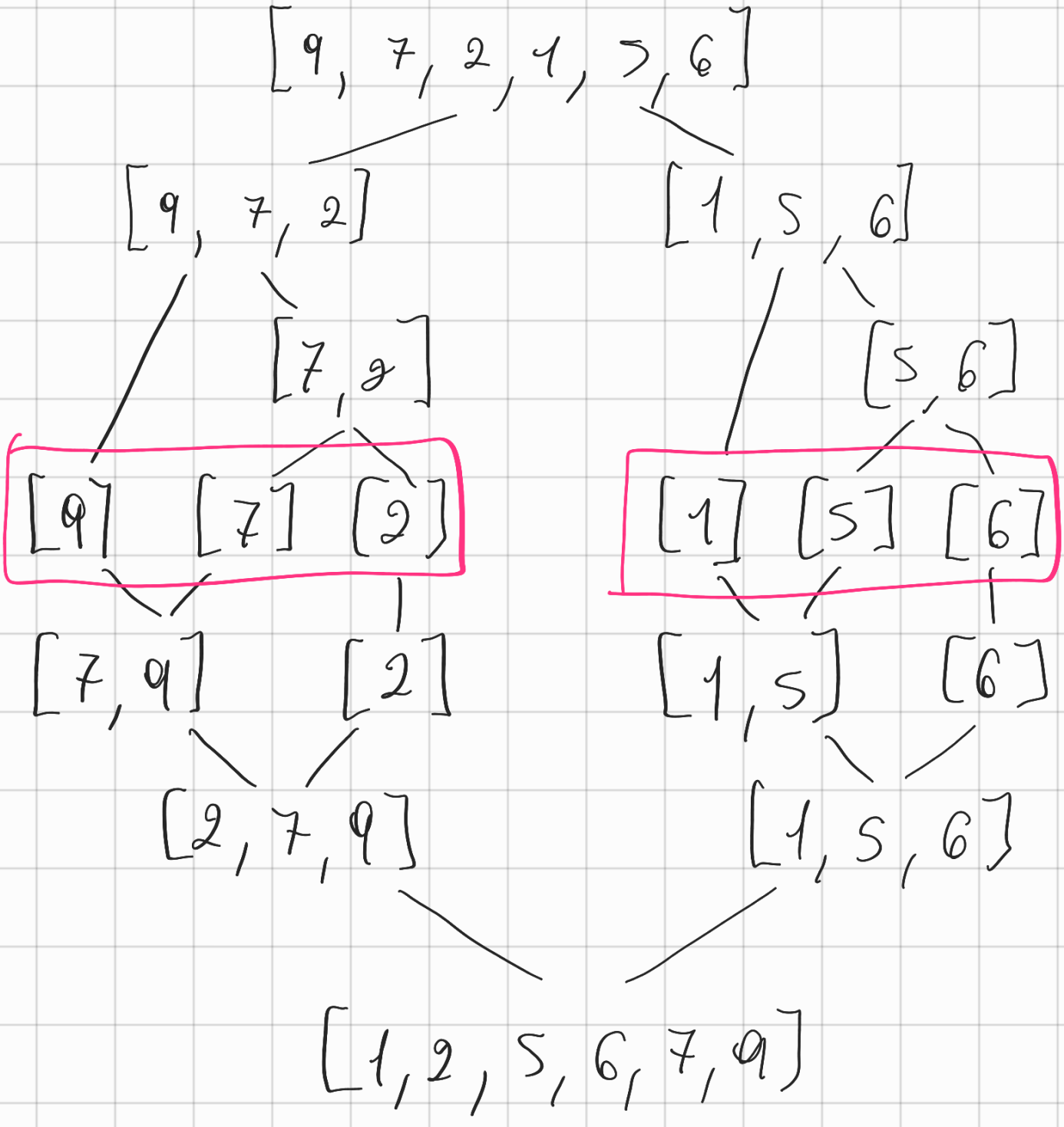
K = 90

23

$5 = 8 + 31 = 39$

| | |
|---|---|
| 19 | 19 |
| 28 | 28 |
| 94 | 94 |
| 23 | 23 |

94
28
19




$O(n \log n) = O(n \log n)$