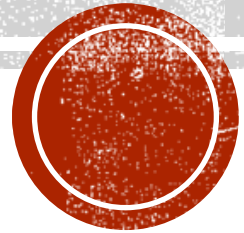
INTRODUÇÃO AO PENSAMENTO ALGORÍTMICO

DIVISÃO E CONQUISTA

Marcela Xavier Ribeiro

DC/UFSCar



DIVISÃO E CONQUISTA

- É preciso revolver um problema com uma entrada grande
- Para facilitar a resolução do problema, a entrada é quebrada em pedaços menores (DIVISÃO)
- Cada pedaço da entrada é então tratado separadamente (CONQUISTA)
- Ao final, os resultados parciais são combinados para gerar o resultado final procurado (COMBINAÇÃO)

DIVISÃO E CONQUISTA

Essa estratégia de projeto de algoritmos consiste no seguinte:

- a instância dada do problema é dividida em duas ou mais instâncias menores,

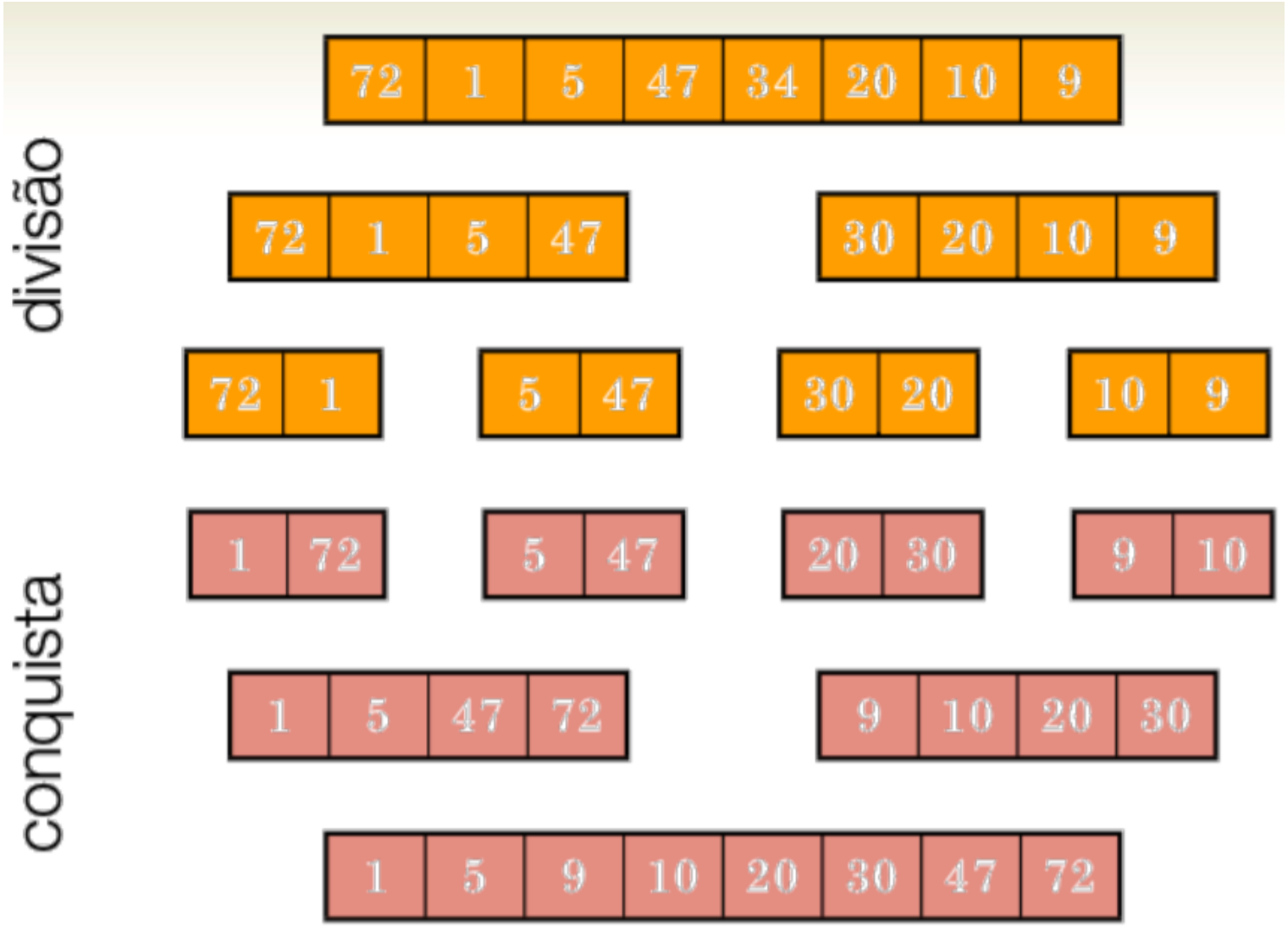
- cada instância menor é resolvida usando o próprio algoritmo que está sendo definido,

- as soluções das instâncias menores são combinadas para produzir uma solução da instância original.

SERÁ QUE JÁ FIZEMOS ISSO?

- Sim!
- Alguns métodos que usam a estratégia de divisão e conquista:
- **Busca binária:** divide a instância em duas menores e resolve uma delas; a fase de combinação é vazia.
- Mergesort;
- Quicksort;.
- ...

MERGESORT - EXEMPLO



Estratégia que pode ser paralelizada: dividir o trabalho entre processadores

Fonte da Figura: http://www3.decom.ufop.br/toffolo

EXEMPLO – COMO FAZER EXPONENCIAL

- Vamos fazer um algoritmo que calcula a exponencial a^n;
- 2^3 = 2*2*2

```
int exp (int a, int n) {
    int p = 1;
    for (int i = 0; i < n; i++)
        p = p * a;
    return p;
}
```

EXPONENCIAL RECURSIVA

```
int exprec(int a, int n) {
    if (n == 0)
        return 1;
     return a* exprec(a,n-1);
}
```

EXPONENCIAL RECURSIVA

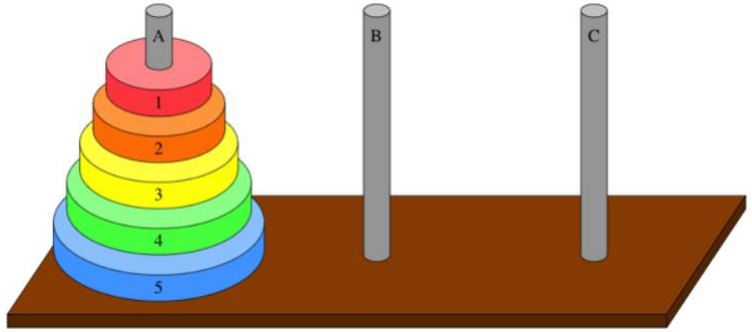
```
int exp2(int a, int n) {
    if (n == 0)
        return 1;
    if (n % 2 == 0)
        return exp2(a,n/2) * exp2 (a,n/2);
    else
        return exp2(a, n/2) * exp2(a,n/2) * a;
}
```

EXPONENCIAL RECURSIVA

```
int exp3(int a, int n) {
    if (n == 0)
        return 1;
    int half = exp3(a,n/2) ;
    int resp = half*half;
    if (n % 2 == 1)
        resp = resp* a;
    return resp;
}
```

ATIVIDADE - TORRE DE HANOI

- A Torre de Hanói consiste em uma base contendo três pinos, em um dos quais são dispostos *n* discos uns sobre os outros, em ordem crescente de diâmetro, de cima para baixo. O problema consiste em passar todos os discos de um pino para último pino, usando um pino auxiliar, de maneira que um disco maior nunca fique em cima de outro menor em nenhuma situação.



Fonte da Figura
https://pt.khanacademy.org/computing/computer-science/algorithms/towers-of-hanoi/a/towers-of-hanoi

- Jogue o jogo:

https://www.somatematica.com.br/jogos/hanoi/

Como computacionalmente podemos encontrar a sequência de passos que pode resolver esse problema?

INTRODUÇÃO AO PENSAMENTO ALGORÍTMICO

DIVISÃO E CONQUISTA

Marcela Xavier Ribeiro

DC/UFSCar