Aula 8

Programação Dinâmica

Exemplo (fibonacci):

$$F(n) = F(n-1) + F(n-2) \;\; ; \qquad F(0) = 1; F(1) = 1$$

Como fazer para calcular F(n)?

Exemplo (fibonacci):

$$F(n) = F(n-1) + F(n-2) \;; \qquad F(0) = 1; F(1) = 1$$

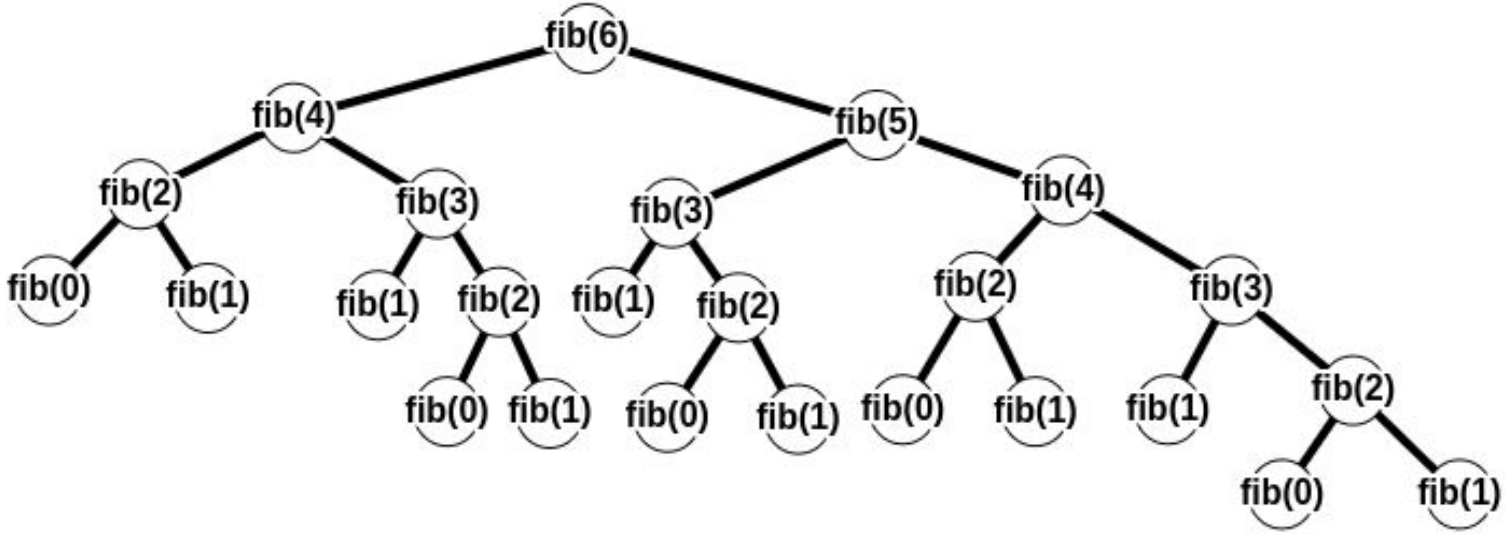
Como fazer para calcular F(n)?

Solução recursiva:

```
int fib(int n){
    if(n == 1 || n == 0) return 1;
    return fib(n-1) + fib(n-2);
}
```

Exemplo (fibonacci):

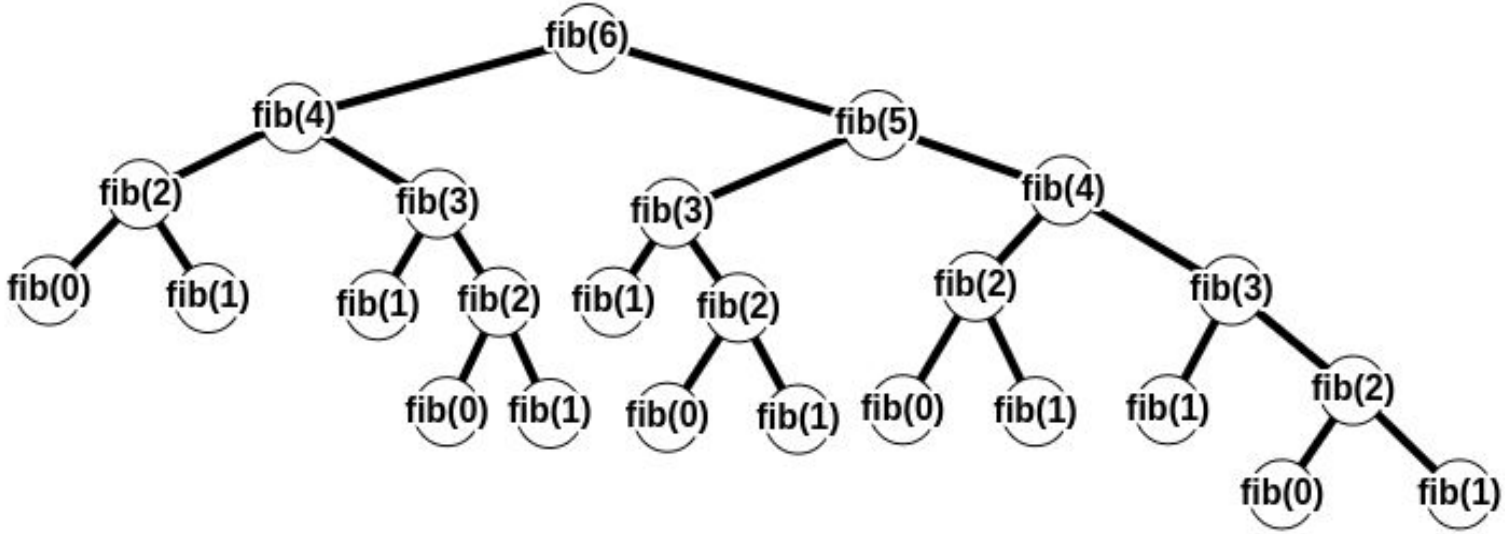
Para calcular F(6):



Exemplo (fibonacci):

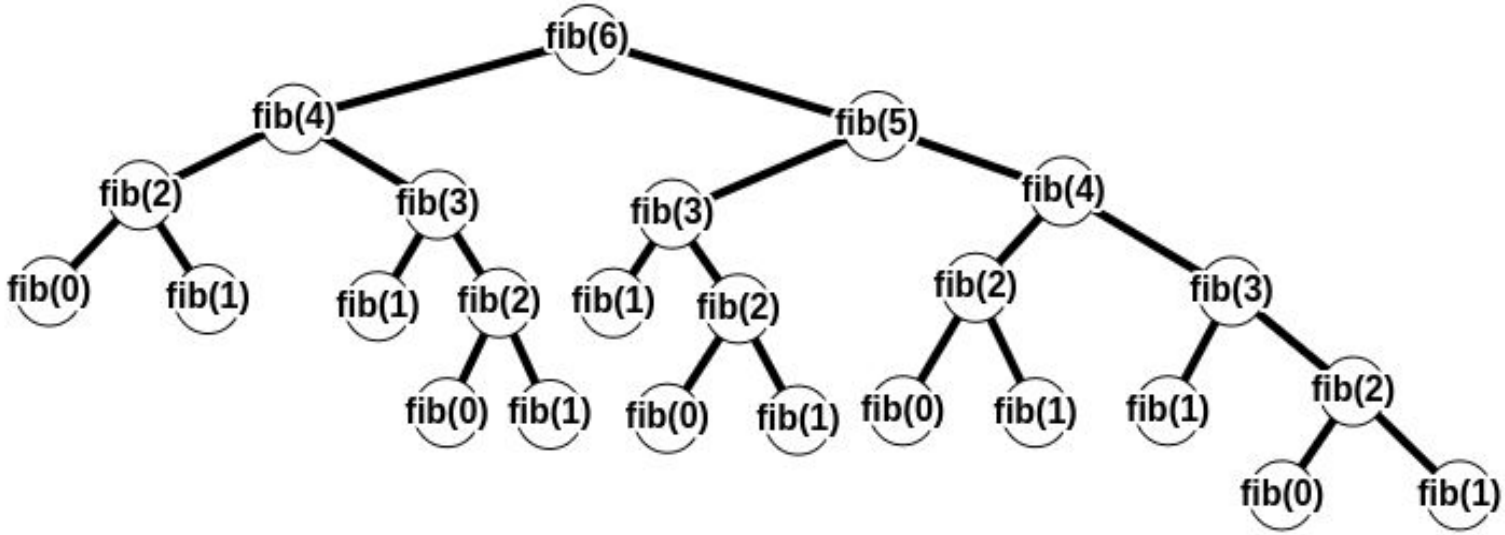
Para calcular F(6):

Quanto tempo demora?



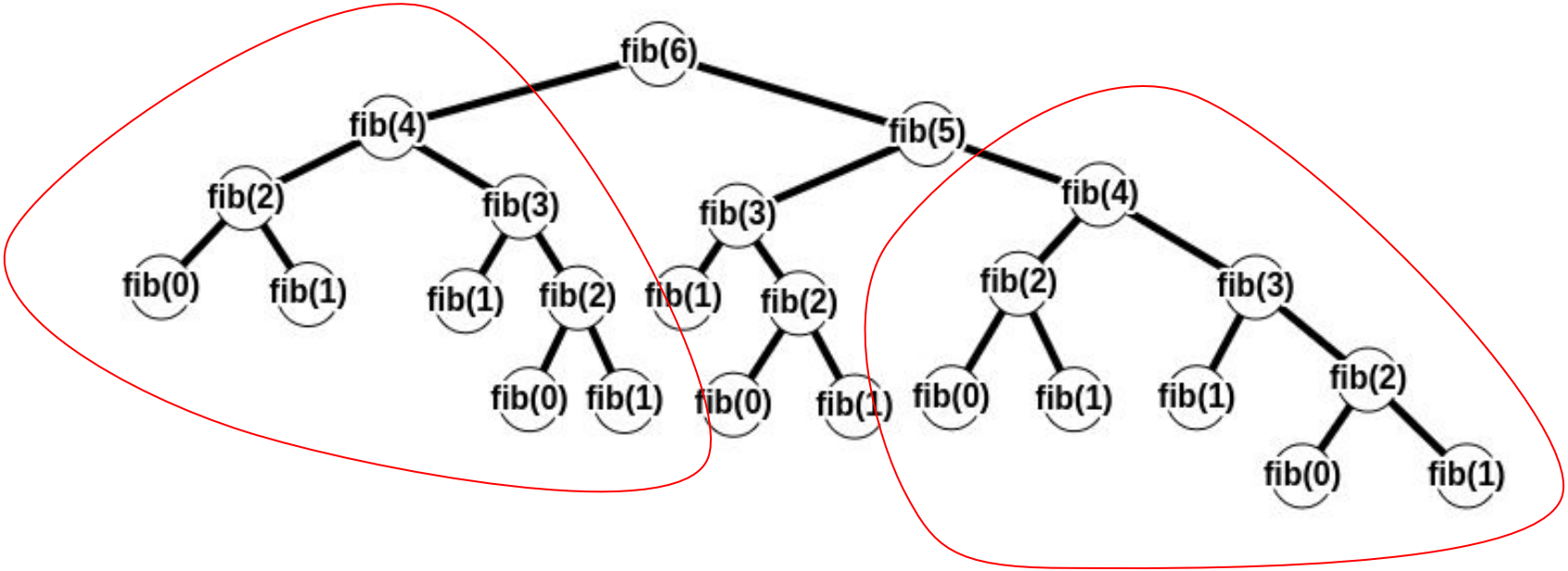
Exemplo (fibonacci):

Para calcular F(6): Quanto tempo demora? exponencial



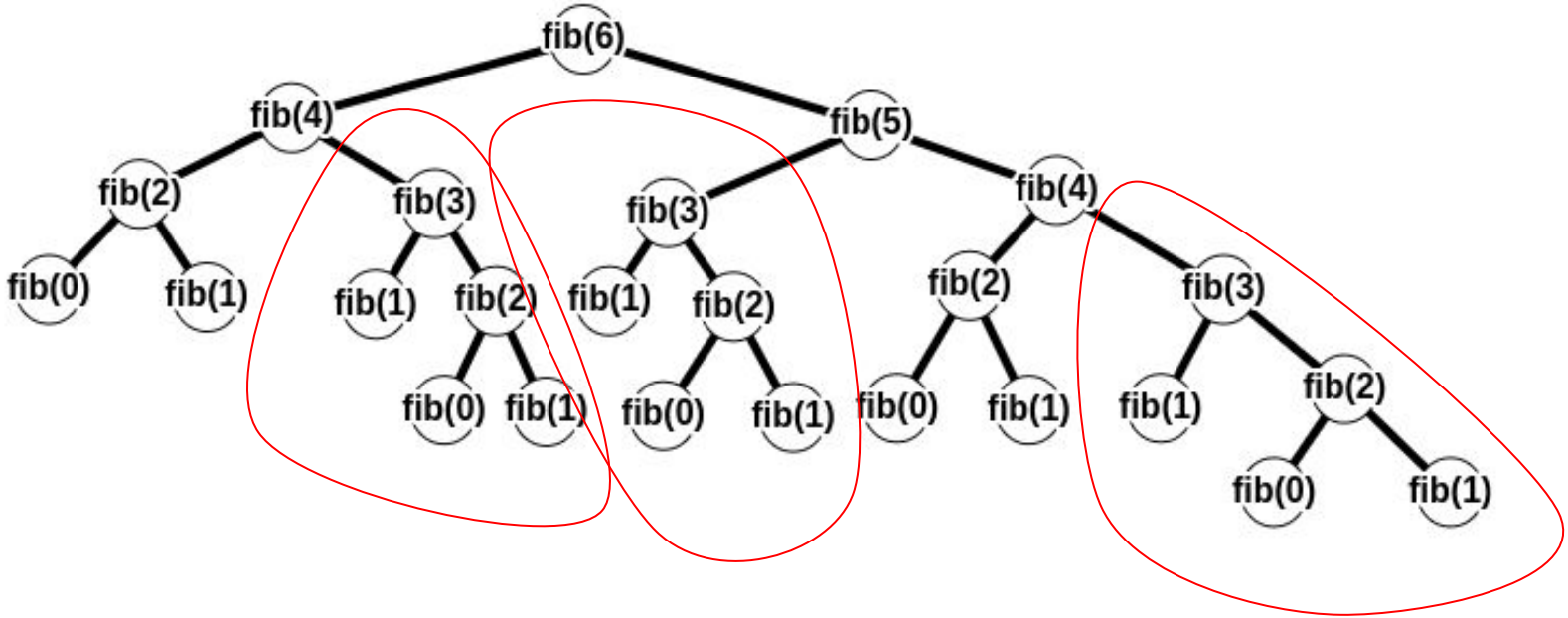
Exemplo (fibonacci):

Para calcular F(6):



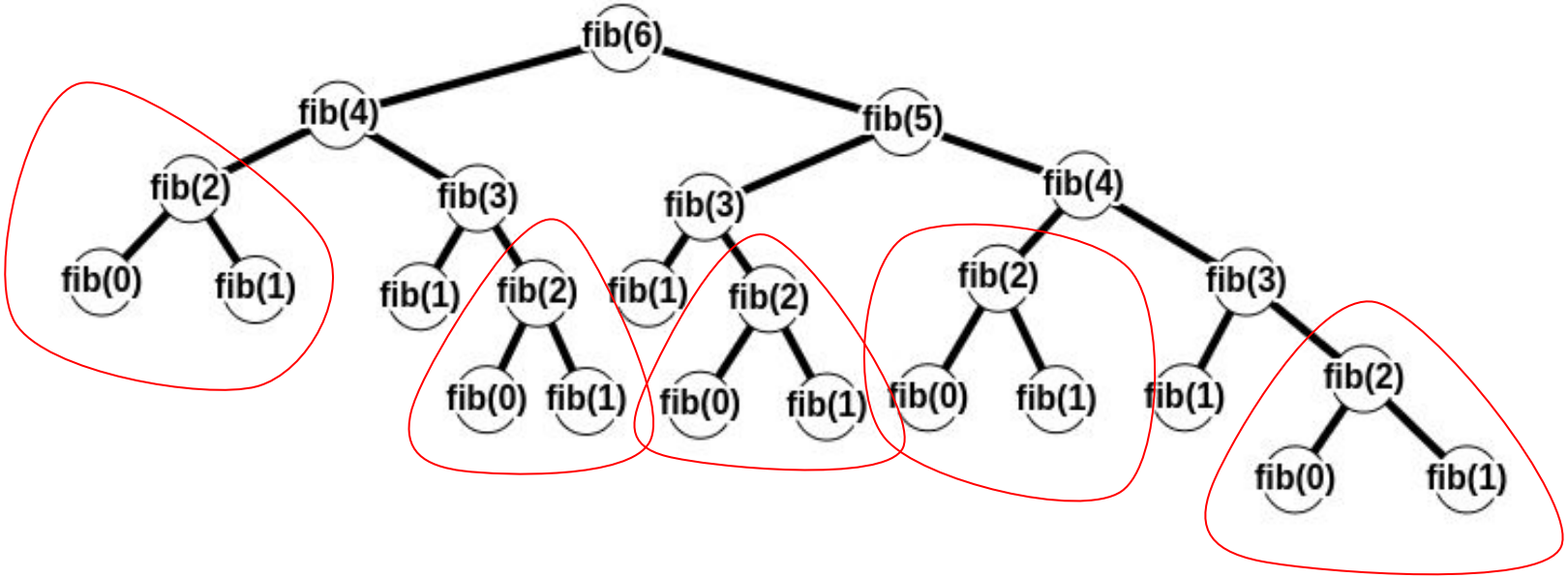
Exemplo (fibonacci):

Para calcular F(6):



Exemplo (fibonacci):

Para calcular F(6):



Exemplo (fibonacci):

Ideia da programação dinâmica: guardar os valores em algum lugar (por exemplo, num array) para não precisar recalcular tudo de novo.

Exemplo (fibonacci):

Ideia da programação dinâmica: guardar os valores em algum lugar (por exemplo, num array) para não precisar recalcular tudo de novo.

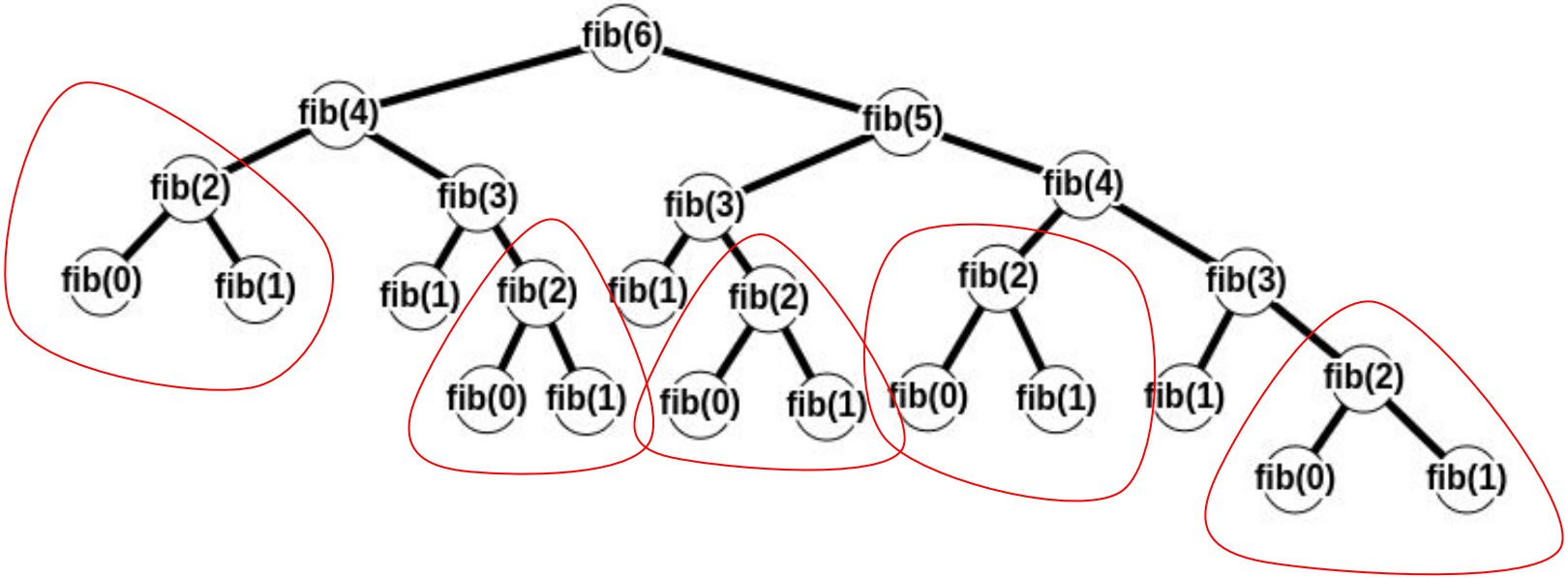
Nova recursão, com programação dinâmica (memoização):

(se memo[i] = -1 então ainda não foi calculado)

```
int fib(int n){
    if(memo[n] == -1) memo[n] = fib(n-1) + fib(n-2);
    return memo[n];
}
```

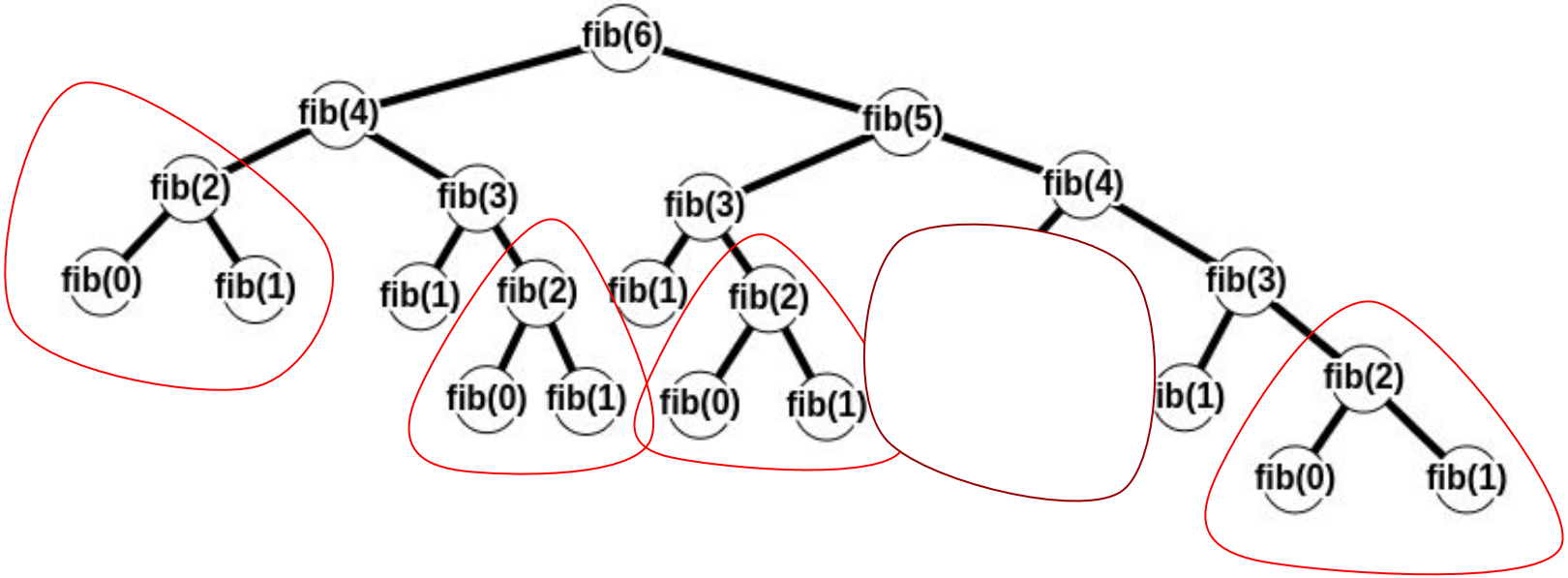
Exemplo (fibonacci):

Para calcular F(6):



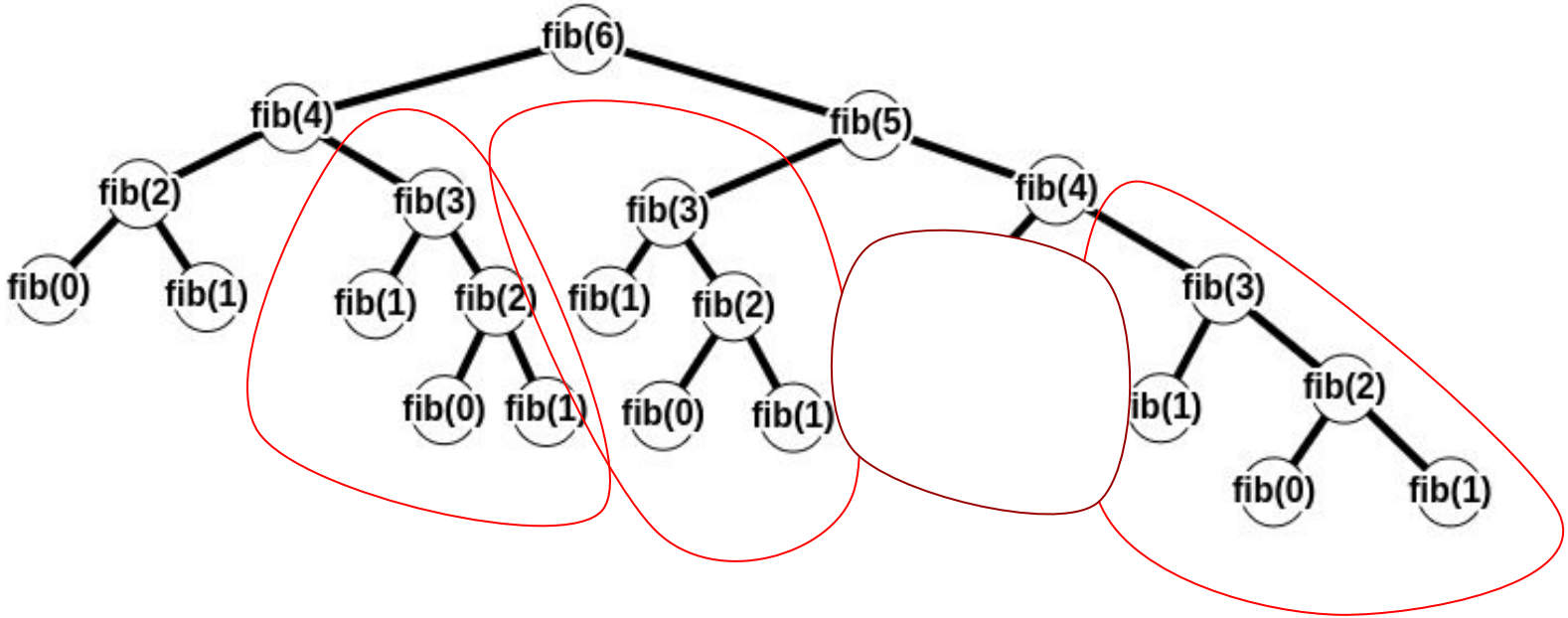
Exemplo (fibonacci):

Para calcular F(6):



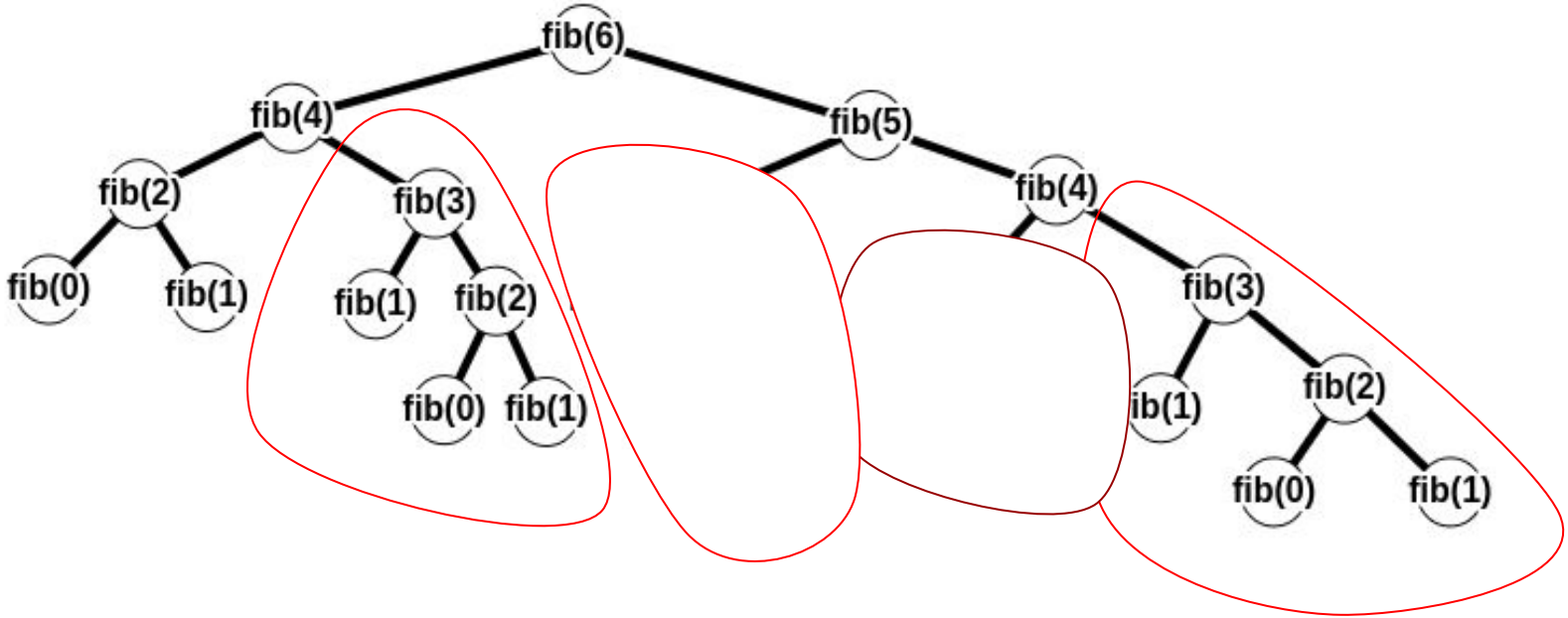
Exemplo (fibonacci):

Para calcular F(6):



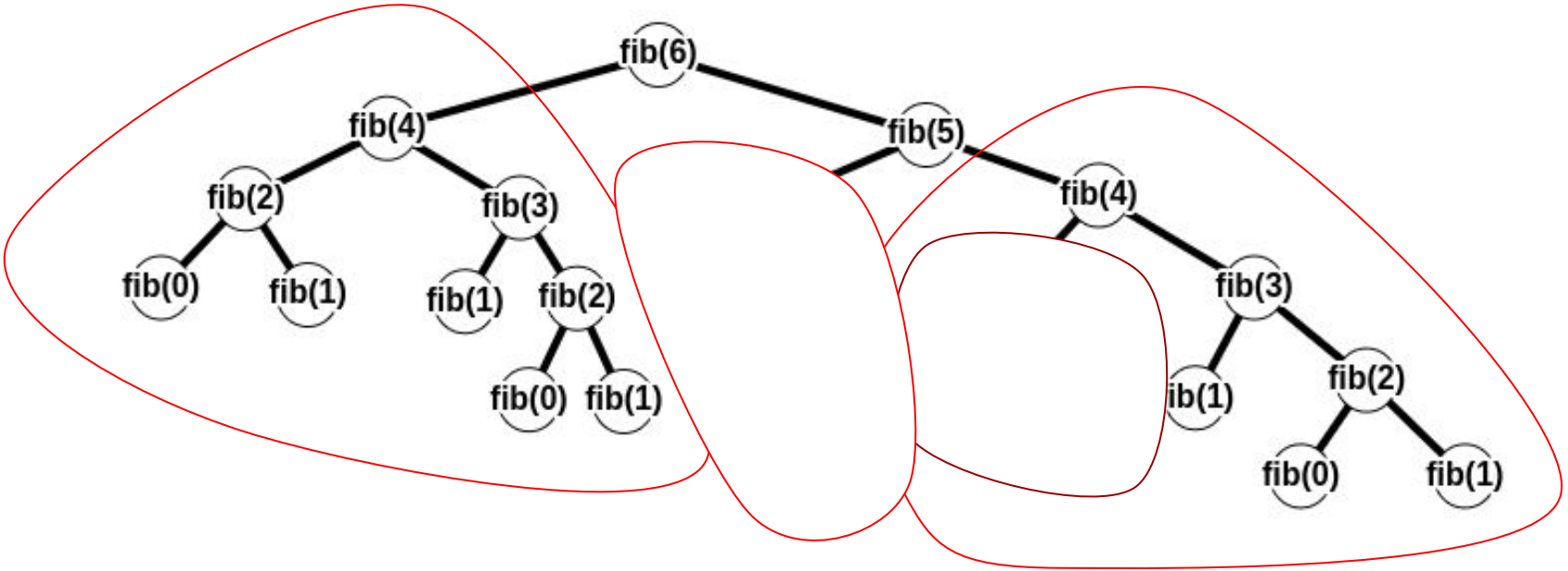
Exemplo (fibonacci):

Para calcular F(6):



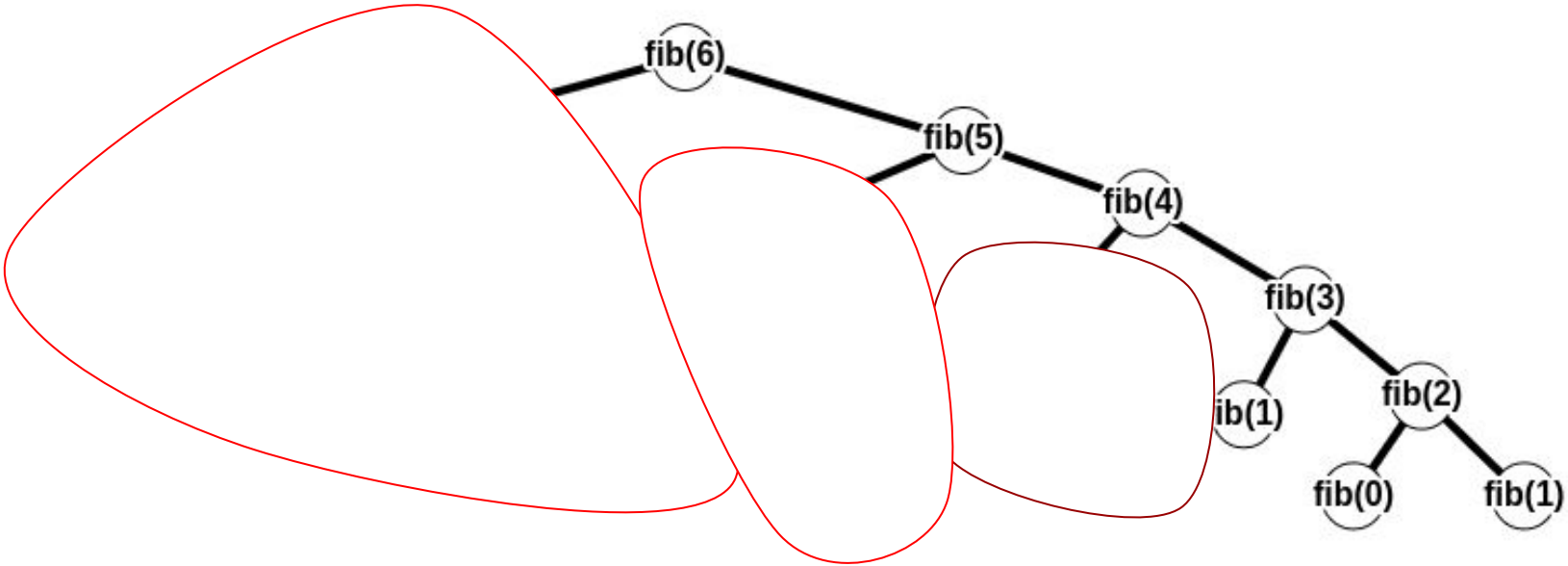
Exemplo (fibonacci):

Para calcular F(6):



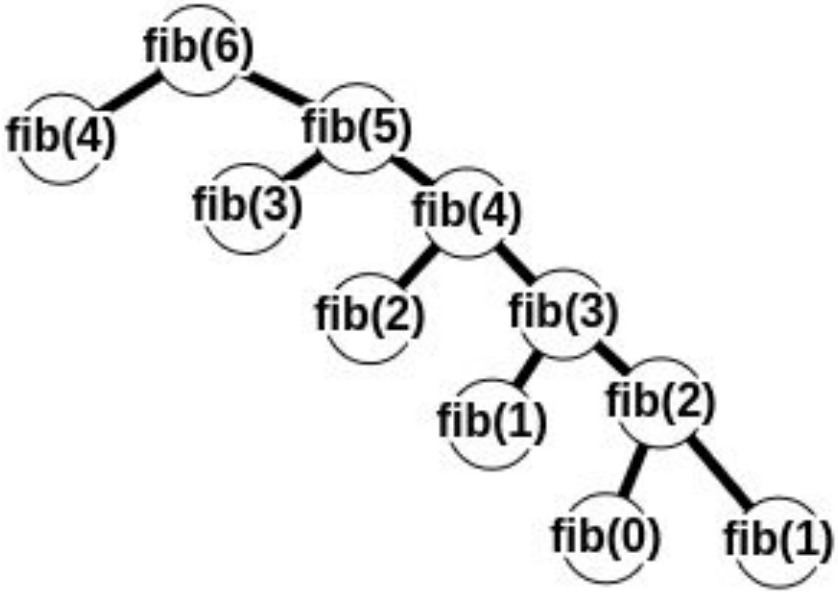
Exemplo (fibonacci):

Para calcular F(6):



Exemplo (fibonacci):

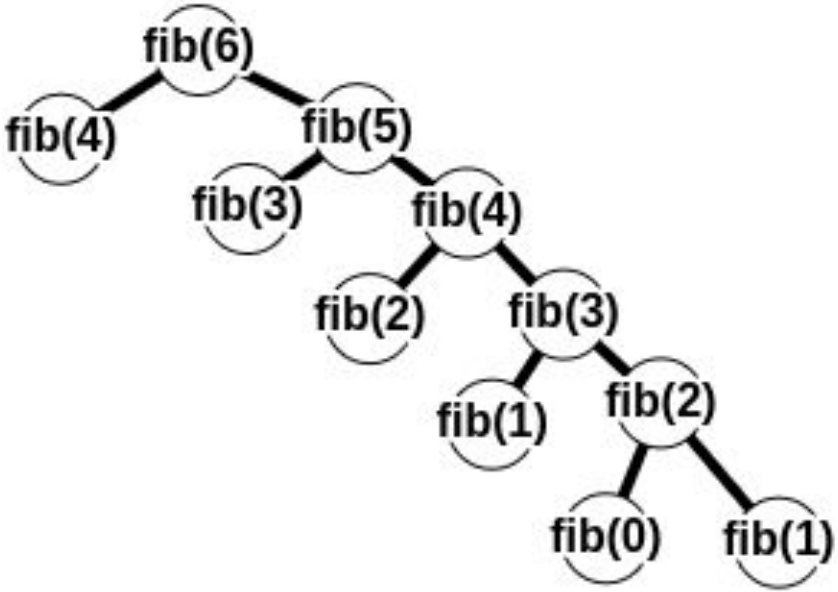
Para calcular F(6):



Exemplo (fibonacci):

Para calcular F(6):

Qual a complexidade?

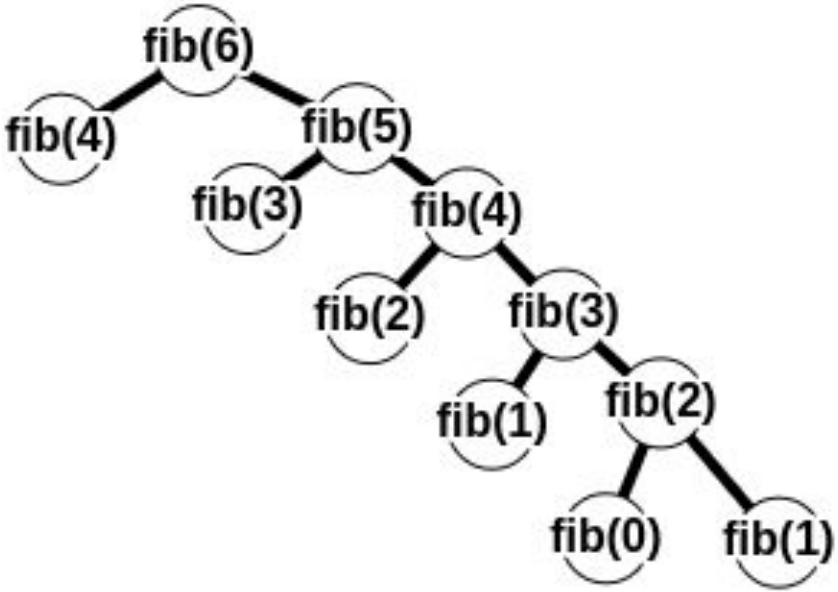


Exemplo (fibonacci):

Para calcular F(6):

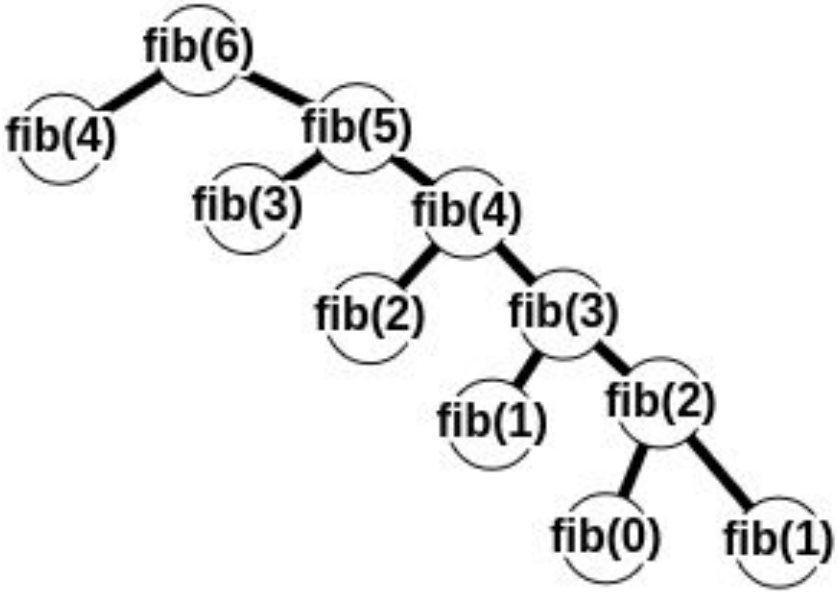
Qual a complexidade?

O(n) !



Exemplo (fibonacci):

Para calcular F(6):



Qual a complexidade?

O(n) !

código.

Exemplo (fibonacci):

Outra forma de calcular fibonacci usando programação dinâmica é fazer de forma iterativa em vez de recursiva.

```
int fib[MAX];
fib[0] = fib[1] = 1;
for(int i = 2; i <= n; i++){
    fib[i] = fib[i-1] + fib[i-2];
}
```

Podemos perceber que fizemos uma troca: gastamos mais memória para gastar menos tempo.

Exemplo (fibonacci):

Mas nem sempre isso é necessário:

```
int ultimo, penultimo, resposta;
resposta = ultimo = penultimo = 1;
for(int i = 2; i <= n; i++){
    resposta = ultimo + penultimo;
    penultimo = ultimo;
    ultimo = resposta;
}
```

Programação Dinâmica

Quando geralmente usamos programação dinâmica?

1 - Problemas de contagem

Programação Dinâmica

Quando geralmente usamos programação dinâmica?

1 - Problemas de contagem

2 - Problemas de max/min

Programação Dinâmica

Quando geralmente usamos programação dinâmica?

1 - Problemas de contagem

2 - Problemas de max/min

3 - Problemas de sim ou não

Programação Dinâmica

Quando geralmente usamos programação dinâmica?

1 - Problemas de contagem

2 - Problemas de max/min

3 - Problemas de sim ou não

(1) lembra combinatória e (2) e (3) lembram gulosos

Usamos PD para problemas com estrutura recursiva

O problema de fibonacci pode ser visto como um problema de contagem.

Programação Dinâmica

Então podemos dizer que PD é uma “recursão com memoização”.

A melhor forma de aprender como usar PD é praticando. Existem alguns tipos de problemas bem conhecidos que usam PD, como Knapsack, Coin Change, LCS , …

Aqui vamos ver o knapsack e alguns outros. Na próxima aula terão mais problemas.

Exemplo (knapsack):

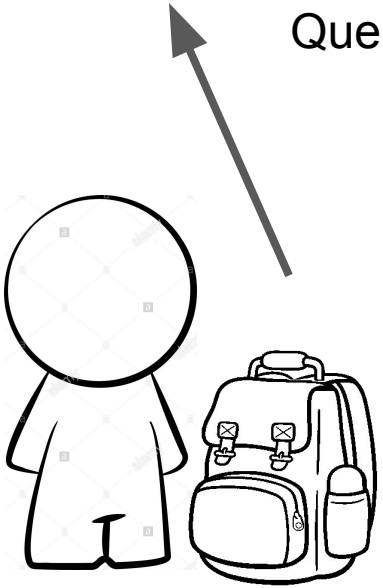
Knapsack é um tipo de problema clássico em programação competitiva, ele pode aparecer de várias maneiras diferentes.

Vamos usar como base o primeiro problema da planilha (que está no spoj):

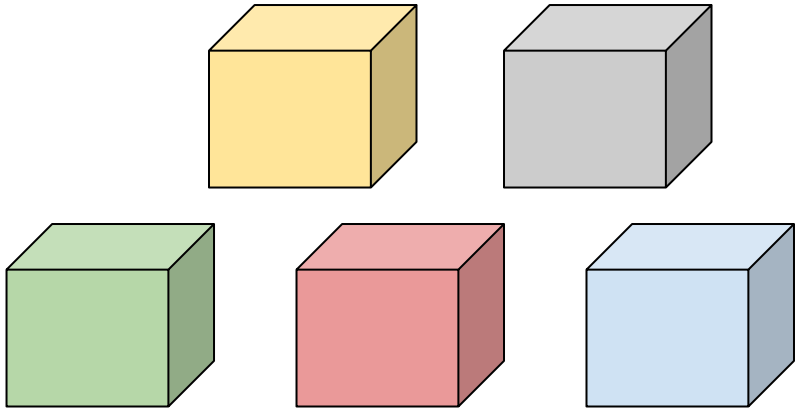
https://www.spoj.com/problems/KNAPSACK/

Exemplo (knapsack):

Queremos saber o máximo valor que conseguimos levar, dado que a mochila tem capacidade S. Temos N objetos, cada um com um peso e valor específico.

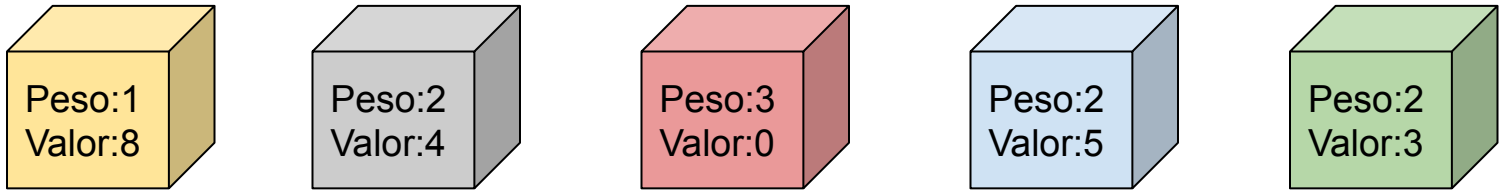


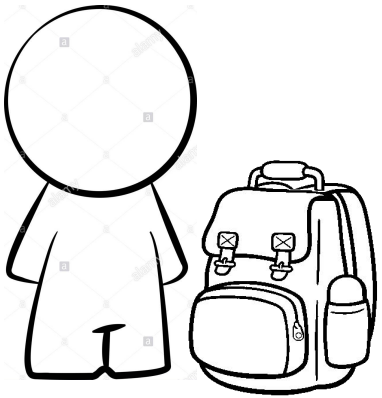
Queremos saber o valor máximo que conseguimos levar.



Exemplo (knapsack):

Caso de exemplo que está no spoj:

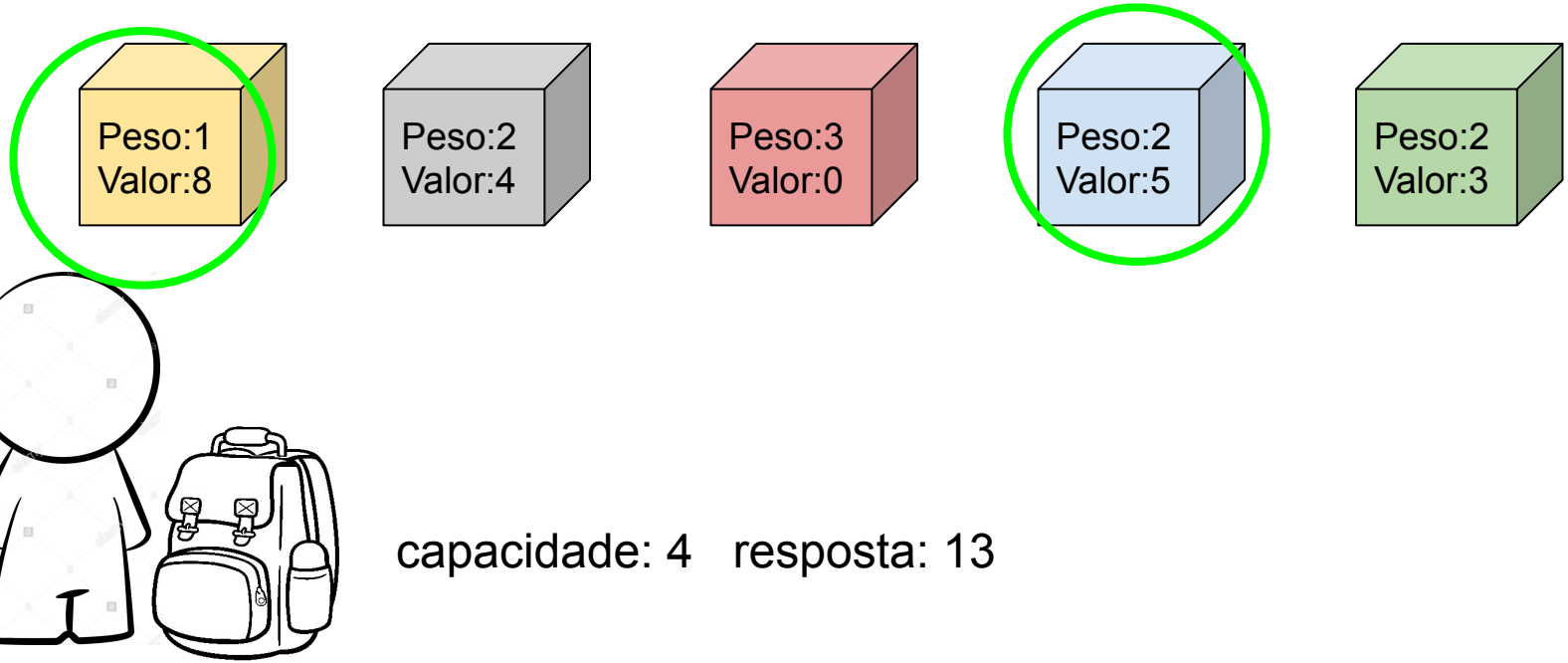




capacidade: 4

Exemplo (knapsack):

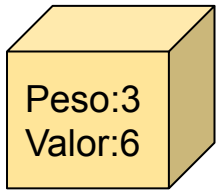
Caso de exemplo que está no spoj:

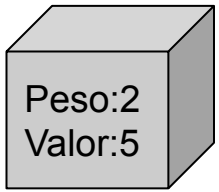


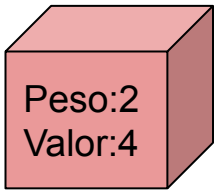
capacidade: 4 resposta: 13

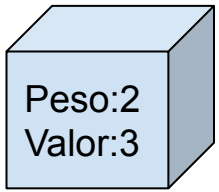
Exemplo (knapsack):

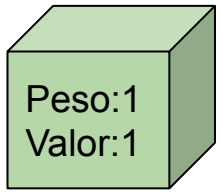
Outro exemplo: (tentativa com guloso: ordena e pega o maior q puder)

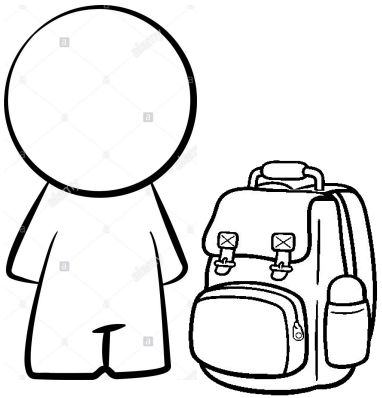








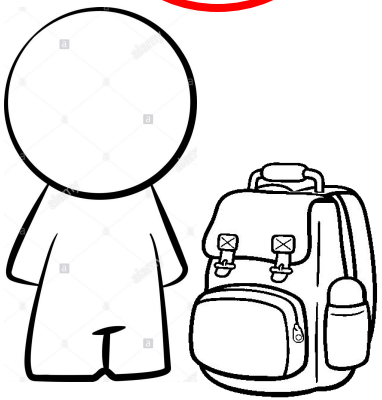




capacidade: 4

Exemplo (knapsack):

Outro exemplo: (tentativa com guloso: ordena e pega o maior q puder)

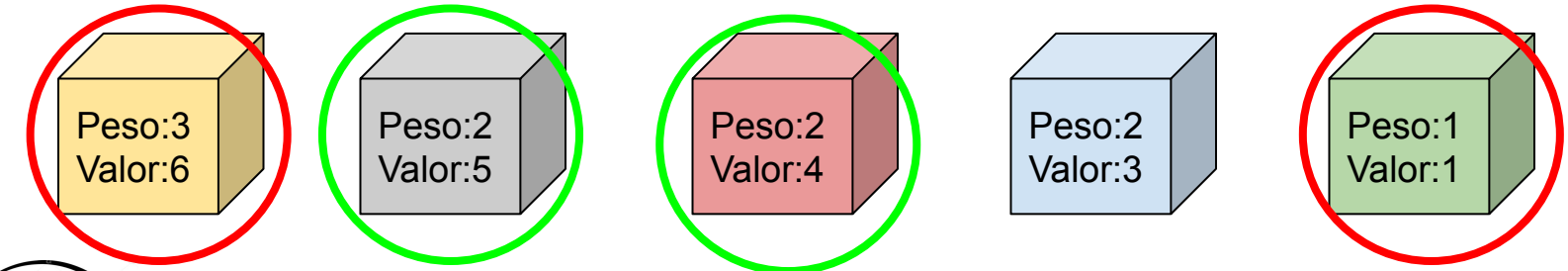


capacidade: 4

resposta: 7?

Exemplo (knapsack):

Outro exemplo:

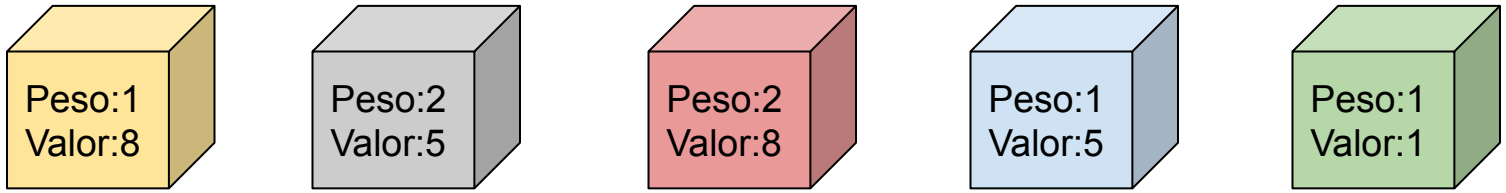


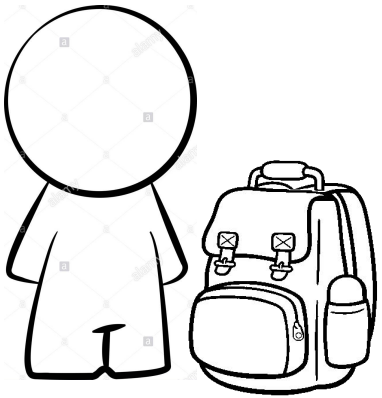
capacidade: 4

resposta: 9

Exemplo (knapsack):

Outro exemplo: (resolvendo na sequência)

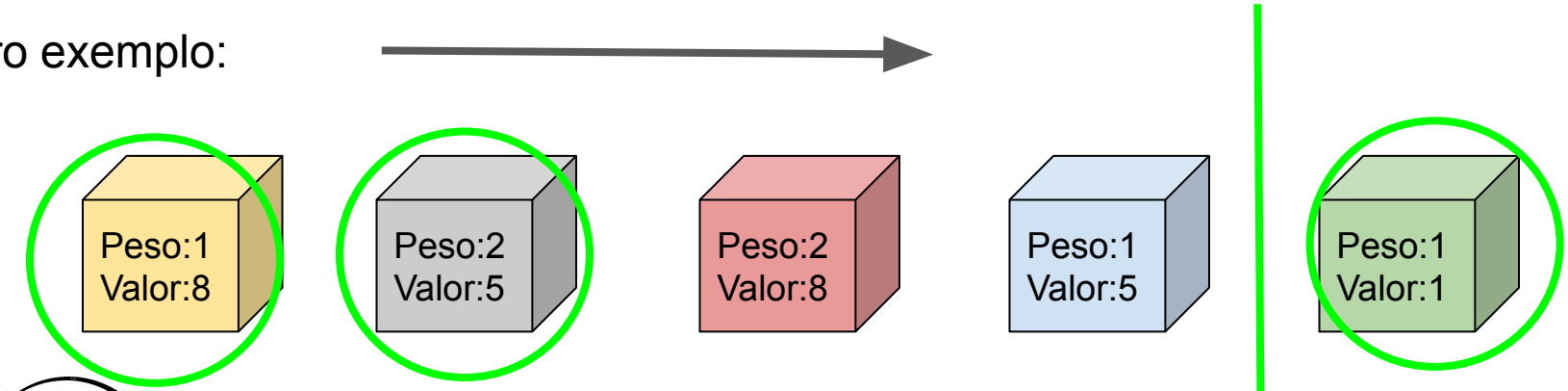


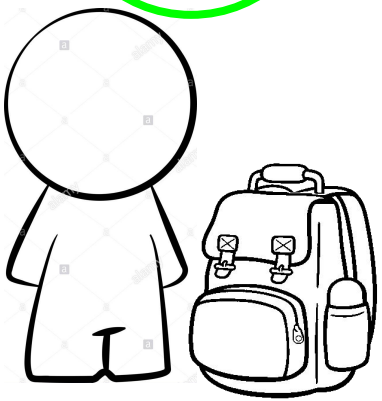


capacidade: 4

Exemplo (knapsack):

Outro exemplo:



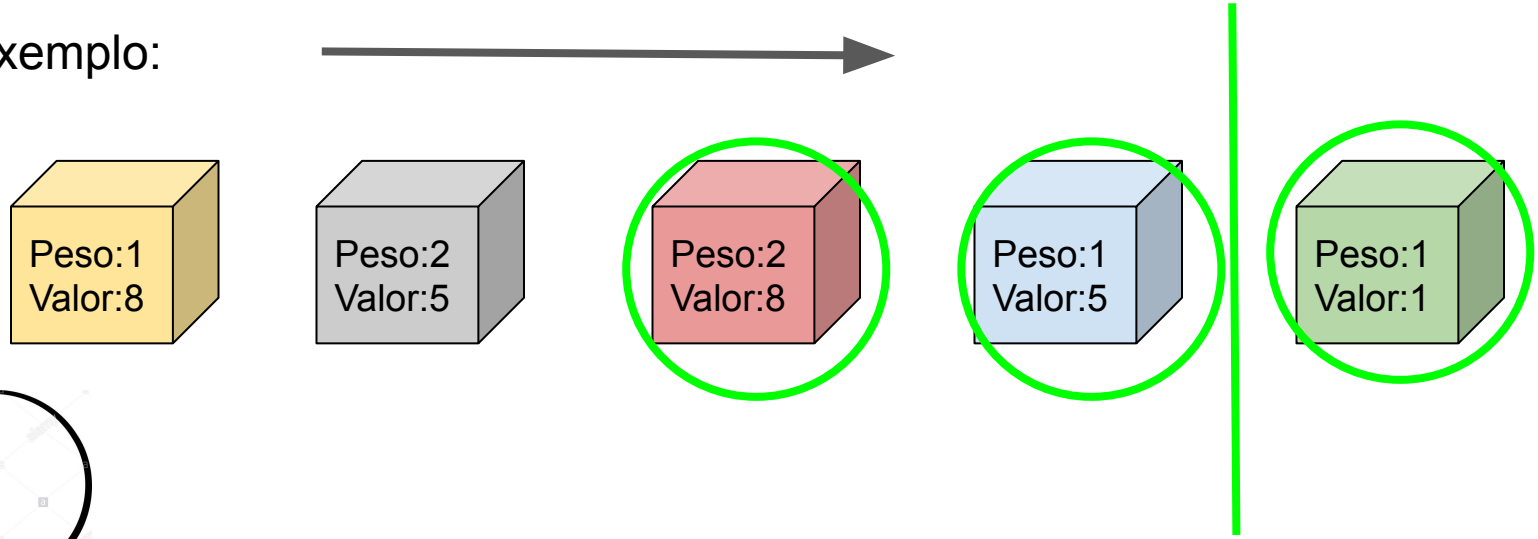


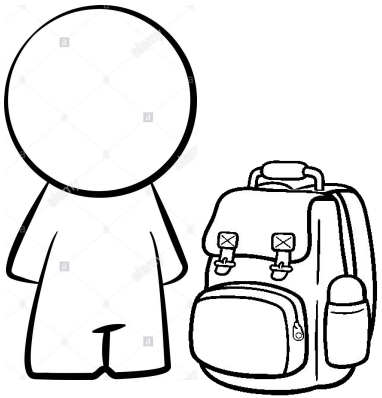
capacidade: 4

resposta: 14

Exemplo (knapsack):

Outro exemplo:





capacidade: 4

resposta: 14

Exemplo (knapsack):

Como podemos resolver esse problema?

Tentamos separar em subproblemas, e para cada um deles procuramos quais informações precisamos ter para resolver o problema. (lembra guloso)

Exemplo (knapsack):

Como podemos resolver esse problema?

Tentamos separar em subproblemas, e para cada um deles procuramos quais informações precisamos ter para resolver o problema. (lembra guloso)

Depois disso uma das formas de resolver o problema é usar uma recursão sem se preocupar muito com a eficiência e depois usar a memoização pra não repetir operações já feitas.

Exemplo (knapsack):

Um jeito de resolver sem se preocupar muito com a eficiência é tentar todas as possibilidades. Para cada um dos objetos podemos escolher coletar ele ou não.

Isso nos dá uma função do tipo pd(índice, total de peso até agr).

Código

Exemplo (knapsack):

Outra maneira de resolver é de forma iterativa:

```
for(int i = 0; i <= n; i++){
        for(int j = 0; j <= s; j++) {
            if(i == 0 || j == 0)
                pd[i][j] = 0;
            else if(p[i - 1] <= j)
                pd[i][j] = max(
                    v[i - 1] + pd[i - 1][j - p[i - 1]],
                    pd[i - 1][j]);
            else
                pd[i][j] = pd[i - 1][j];
        }
    }
```

Exemplo (knapsack):

Além disso, geralmente existem duas formas de pensar no seu algoritmo: bottom-up e top-down (ver diferenças no caso do fibonacci e no knapsack).

Exemplo (knapsack):

Além disso, geralmente existem duas formas de pensar no seu algoritmo: bottom-up e top-down (ver diferenças no caso do fibonacci e no knapsack).

Geralmente para resolver uma pd a dificuldade está em modelar, isto é, saber dividir em subproblemas e descobrir quais são os parâmetros da nossa pd, quais informações a gente precisa em cada etapa. É o que fazemos para resolver problemas com estrutura recursiva.

Exemplo (knapsack):

Além disso, geralmente existem duas formas de pensar no seu algoritmo: bottom-up e top-down (ver diferenças no caso do fibonacci e no knapsack).

Existe um variante legal desse problema em que além de dizer qual o valor máximo também temos que responder quais objetos foram escolhidos.

Aula do ano passado mostra como resolver esse problema:
https://www.youtube.com/watch?v=IHuZAgKGM6Y&t=262s

Exemplo (LCS):

LCS vem de Longest Common Subsequence e como o nome sugere o problema é achar a maior subsequência comum entre duas sequências (por exemplo, uma sequência de ‘chars’).

Ex:

1 2 3 4 5 6

1 4 2 3 4 7

Para esse exemplo, vamos resolver https://neps.academy/problem/267 do Neps.

Exemplo (LCS):

Como achar a maior subsequência crescente de **5 8 1 6 7 23 6** ?

Exemplo (LCS):

Como achar a maior subsequência crescente de **5 8 1 6 7 23 6** ?

primeiro ordenamos: **1 5 6 6 7 8 23**

Depois fazemos um LCS entre essa sequência e a primeira.

Exemplo (LCS):

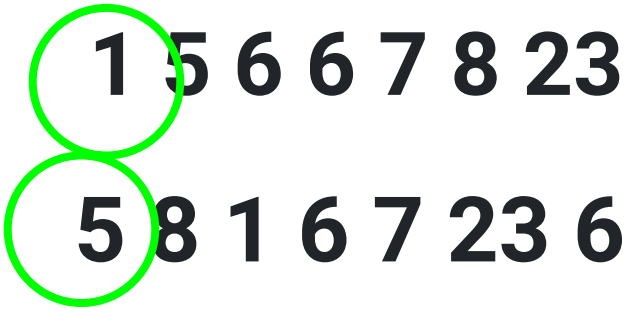
Como achar a maior subsequência crescente de **5 8 1 6 7 23 6** ?

1 5 6 6 7 8 23

5 8 1 6 7 23 6

Exemplo (LCS):

Como achar a maior subsequência crescente de **5 8 1 6 7 23 6** ?



Exemplo (LCS):

Como achar a maior subsequência crescente de **5 8 1 6 7 23 6** ?

1 5 6 6 7 8 23

5 8 1 6 7 23 6

Exemplo (LCS):

Como achar a maior subsequência crescente de **5 8 1 6 7 23 6** ?

1 5 6 6 7 8 23

5 8 1 6 7 23 6

Exemplo (LCS):

Como achar a maior subsequência crescente de **5 8 1 6 7 23 6** ?

1 5 6 6 7 8 23

5 8 1 6 7 23 6

Exemplo (LCS):

código.

Exemplo (Troco):

Problema “Véi, Dá Meu Troco!” do Neps:

https://neps.academy/problem/308

Exemplo (Troco):

TROCO: 19

MOEDAS: 2 5 7

Exemplo (Troco):

TROCO: 19

MOEDAS: 2 5 7

De quantas maneiras podemos chegar no valor 0?

Exemplo (Troco):

TROCO: 19

MOEDAS: 2 5 7

De quantas maneiras podemos chegar no valor 0? dp[0] = 1

Exemplo (Troco):

TROCO: 19

MOEDAS: 2 5 7

dp[1] = 0

Exemplo (Troco):

TROCO: 19

MOEDAS: 2 5 7

dp[1] = 0

dp[2] = 1 = dp[2-2] + 1 = dp[0] + 1 = 1;

… moeda x: dp[i] += dp[i-x];