UFMG

Static Defragmentation

Compilers Laboratory

Elisa Fröhlich
elisa.frohlich@dcc.ufmg.br

Introdução

DCC

Introdução

DCC

MLIR é um compilador para tensores

Introdução

DCC

MLIR é um compilador para tensores que utiliza dialetos

Introdução

DCC

MLIR é um compilador para tensores que utiliza dialetos em diferentes níveis de abstração.

Introdução

DCC

MLIR é um compilador para tensores que utiliza dialetos em diferentes níveis de abstração. Esses tensores têm que ser alocados

Introdução

DCC

MLIR é um compilador para tensores que utiliza dialetos em diferentes níveis de abstração. Esses tensores têm que ser alocados como buffers na memória.

Introdução

DCC

MLIR é um compilador para tensores que utiliza dialetos em diferentes níveis de abstração. Esses tensores têm que ser alocados como buffers na memória.

```
func.func @tf2onnx(%arg0: !torch.vtensor<[?,32,32,3],f32>) -> !torch.vtensor<[?,10],f32> {
    ...
    %7 = torch.operator "onnx.Transpose"(%arg0) : (!torch.vtensor<[?,32,32,3],f32>) ->
!torch.vtensor<[?,3,32,32],f32>
    ...
}
```

Dialeto torch

Introdução

DCC

MLIR é um compilador para tensores que utiliza dialetos em diferentes níveis de abstração. Esses tensores têm que ser alocados como buffers na memória.

```
func.func @tf2onnx(%arg0: !torch.vtensor<[?,32,32,3],f32>) -> !torch.vtensor<[?,10],f32> {
    ...
    %7 = torch.operator "onnx.Transpose"(%arg0) : (!torch.vtensor<[?,32,32,3],f32>) ->
!torch.vtensor<[?,3,32,32],f32>
    ...
}
```

Dialeto linalg

```
func.func @tf2onnx(%arg0: tensor<?x32x32x3xf32>) -> tensor<?x10xf32> {
    %c0 = arith.constant 0 : index
    ...
    %dim = tensor.dim %arg0, %c0 : tensor<?x32x32x3xf32>
    %0 = tensor.empty(%dim) : tensor<?x3x32x32xf32>
    %transposed = linalg.transpose ins(%arg0 : tensor<?x32x32x3xf32>) outs(%0 : tensor<?x3x32x32xf32>)
    ...
}
```

Introdução

DCC

MLIR é um compilador para tensores que utiliza dialetos em diferentes níveis de abstração. Esses tensores têm que ser alocados como buffers na memória.

```
func.func @tf2onnx(%arg0: !torch.vtensor<[?,32,32,3],f32>) -> !torch.vtensor<[?,10],f32> {
    ...
    %7 = torch.operator "onnx.Transpose"(%arg0) : (!torch.vtensor<[?,32,32,3],f32>) ->
!torch.vtensor<[?,3,32,32],f32>
    ...
}
```

Dialeto memref

```
func.func @tf2onnx(%arg0: memref<?x32x32x3xf32>) -> memref<?x10xf32> {
    %c0 = arith.constant 0 : index
    %dim = memref.dim %arg0, %c0 : memref<?x32x32x3xf32>
    %alloc = memref.alloc(%dim) {alignment = 64 : i64} : memref<?x3x32x32xf32>
    linalg.transpose ins(%arg0 : memref<?x32x32x3xf32>) outs(%alloc : memref<?x3x32x32xf32>)
    ...
    memref.dealloc %alloc : memref<?x3x32x32xf32>
    ...
}
```

Introdução

DCC

MLIR é um compilador para tensores que utiliza dialetos em diferentes níveis de abstração. Esses tensores têm que ser alocados como buffers na memória.

```
func.func @tf2onnx(%arg0: !torch.vtensor<[?,32,32,3],f32>) -> !torch.vtensor<[?,10],f32> {
    ...
    %7 = torch.operator "onnx.Transpose"(%arg0) : (!torch.vtensor<[?,32,32,3],f32>) ->
!torch.vtensor<[?,3,32,32],f32>
    ...
}
```

Dialeto LLVM

```
llvm.func @tf2onnx(...) -> !llvm.struct<(ptr, ptr, i64, array<2 x i64>, array<2 x i64>)>
{
    ...
    %197 = llvm.call @malloc(%196) : (i64) -> !llvm.ptr
    // Algumas instruções para calcular a transposta da matriz
    %217 = llvm.insertvalue %188, %216[4, 3] : !llvm.struct<(ptr, ptr, i64, array<4 x i64>, array<4 x i64>)>
    ...
    %2144 = llvm.extractvalue %217[0] : !llvm.struct<(ptr, ptr, i64, array<4 x i64>, array<4 x i64>)>
    llvm.call @free(%2144) : (!llvm.ptr) -> ()
    ...
}
```

Dynamic Storage Allocation

DCC

Dynamic Storage Allocation

DCC

Dado um conjunto de buffers,

Dynamic Storage Allocation

DCC

Dado um conjunto de buffers, com tamanho e tempo de alocação e desalocação

Dynamic Storage Allocation

DCC

Dado um conjunto de buffers, com tamanho e tempo de alocação e desalocação conhecidos para cada buffer,

Dynamic Storage Allocation

DCC

Dado um conjunto de buffers, com tamanho e tempo de alocação e desalocação conhecidos para cada buffer, queremos encontrar uma alocação

Dynamic Storage Allocation

DCC

Dado um conjunto de buffers, com tamanho e tempo de alocação e desalocação conhecidos para cada buffer, queremos encontrar uma alocação para esses buffers

Dynamic Storage Allocation

DCC

Dado um conjunto de buffers, com tamanho e tempo de alocação e desalocação conhecidos para cada buffer, queremos encontrar uma alocação para esses buffers que minimize a quantidade de memória.

Dynamic Storage Allocation

DCC

Dado um conjunto de buffers, com tamanho e tempo de alocação e desalocação conhecidos para cada buffer, queremos encontrar uma alocação para esses buffers que minimize a quantidade de memória. Esse problema é conhecido na literatura

Dynamic Storage Allocation

DCC

Dado um conjunto de buffers, com tamanho e tempo de alocação e desalocação conhecidos para cada buffer, queremos encontrar uma alocação para esses buffers que minimize a quantidade de memória. Esse problema é conhecido na literatura como Dynamic Storage Allocation (DSA)

Dynamic Storage Allocation

DCC

Dado um conjunto de buffers, com tamanho e tempo de alocação e desalocação conhecidos para cada buffer, queremos encontrar uma alocação para esses buffers que minimize a quantidade de memória. Esse problema é conhecido na literatura como Dynamic Storage Allocation (DSA), e é um problema NP-Completo.

Exemplo

DCC

Exemplo

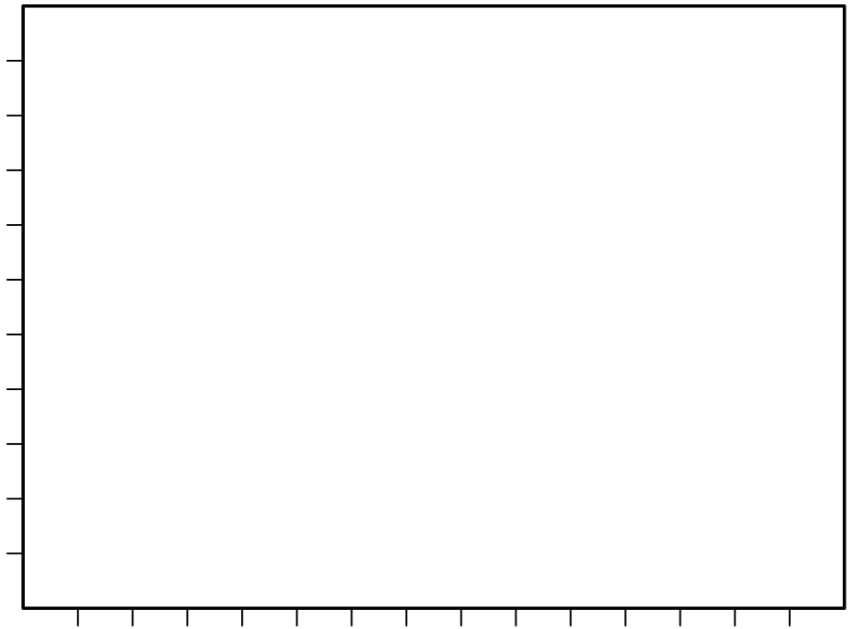
DCC

```
B1 = (2, 6, 2)
B2 = (3, 8, 3)
B3 = (4, 10, 1)
B4 = (6, 11, 4)
B5 = (9, 13, 3)
B6 = (11, 13, 1)
```

Exemplo

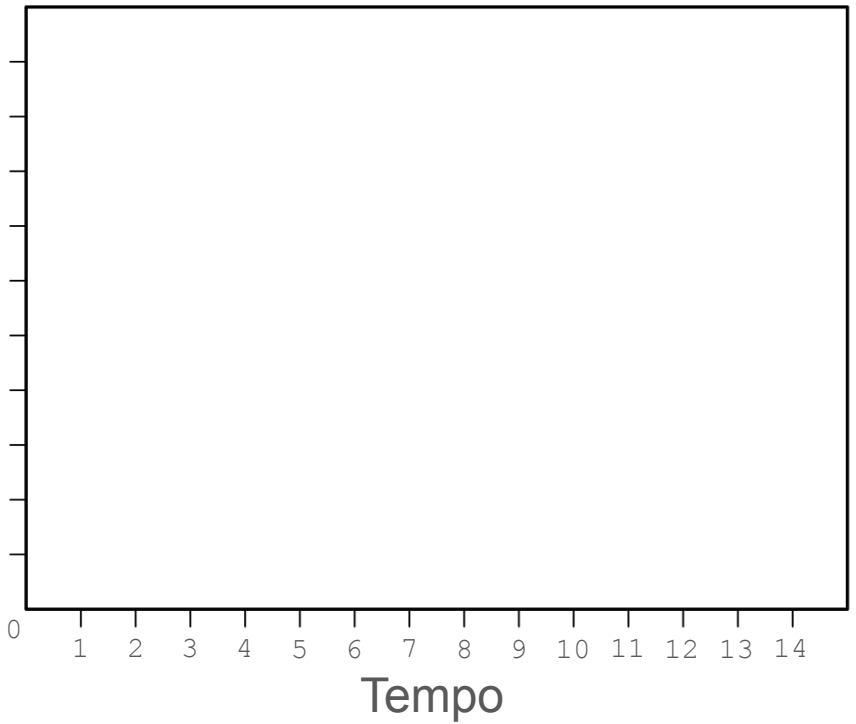
DCC

```
B1 = (2, 6, 2)
B2 = (3, 8, 3)
B3 = (4, 10, 1)
B4 = (6, 11, 4)
B5 = (9, 13, 3)
B6 = (11, 13, 1)
```



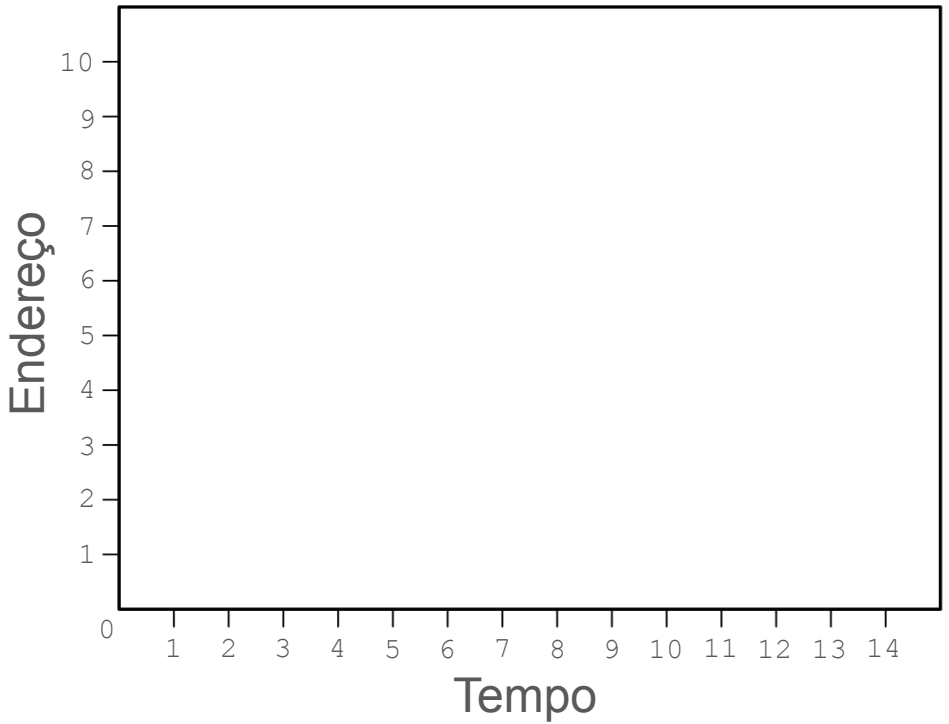
Exemplo

```
B1 = (2, 6, 2)
B2 = (3, 8, 3)
B3 = (4, 10, 1)
B4 = (6, 11, 4)
B5 = (9, 13, 3)
B6 = (11, 13, 1)
```



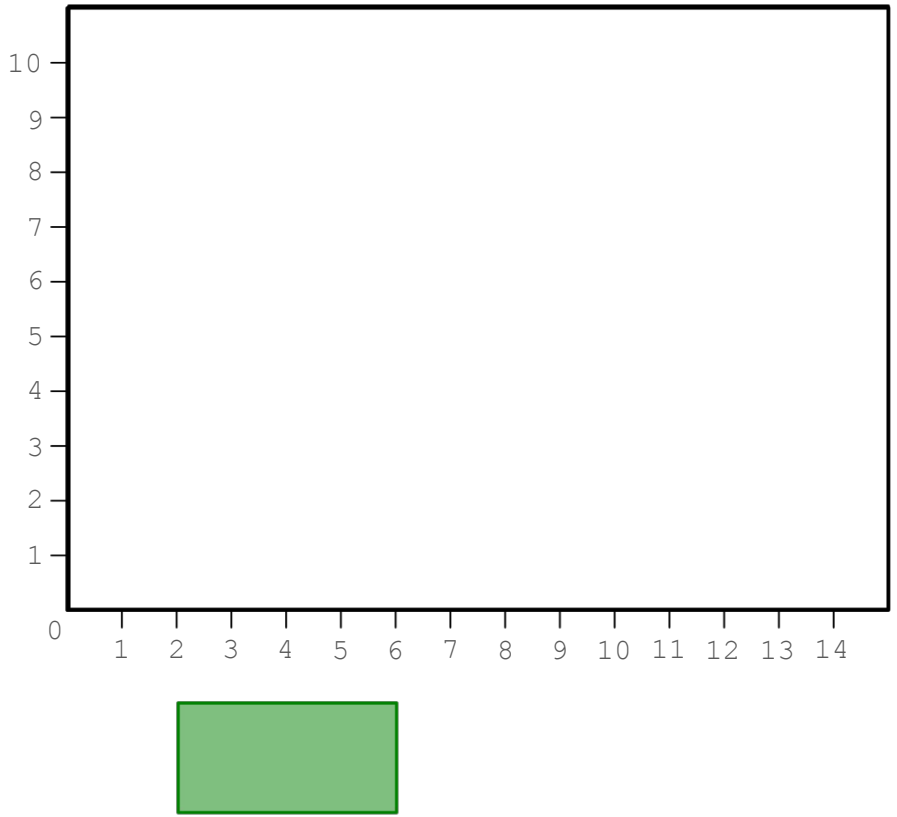
Exemplo

```
B1 = (2, 6, 2)
B2 = (3, 8, 3)
B3 = (4, 10, 1)
B4 = (6, 11, 4)
B5 = (9, 13, 3)
B6 = (11, 13, 1)
```



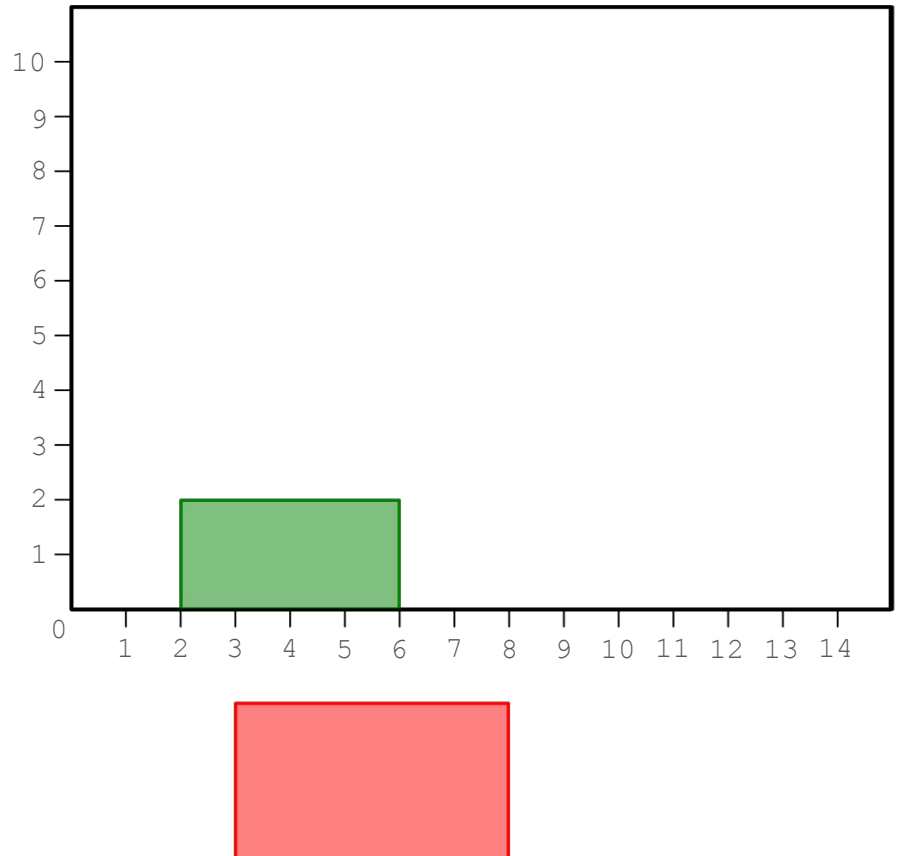
Exemplo

```
B1 = (2, 6, 2)
B2 = (3, 8, 3)
B3 = (4, 10, 1)
B4 = (6, 11, 4)
B5 = (9, 13, 3)
B6 = (11, 13, 1)
```



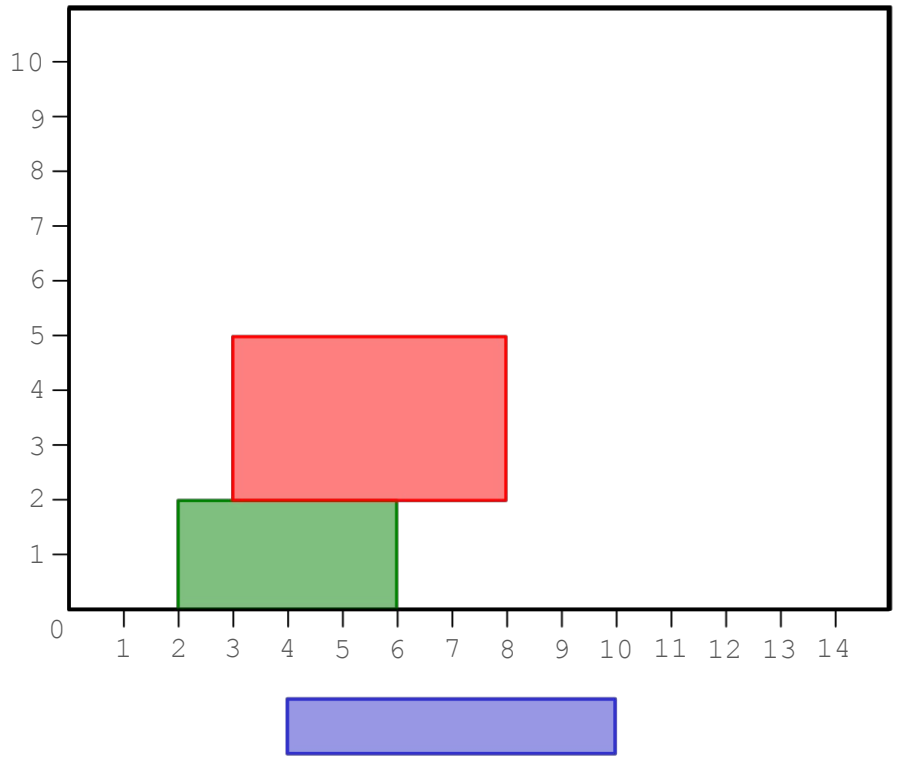
Exemplo

```
B1 = (2, 6, 2)
B2 = (3, 8, 3)
B3 = (4, 10, 1)
B4 = (6, 11, 4)
B5 = (9, 13, 3)
B6 = (11, 13, 1)
```



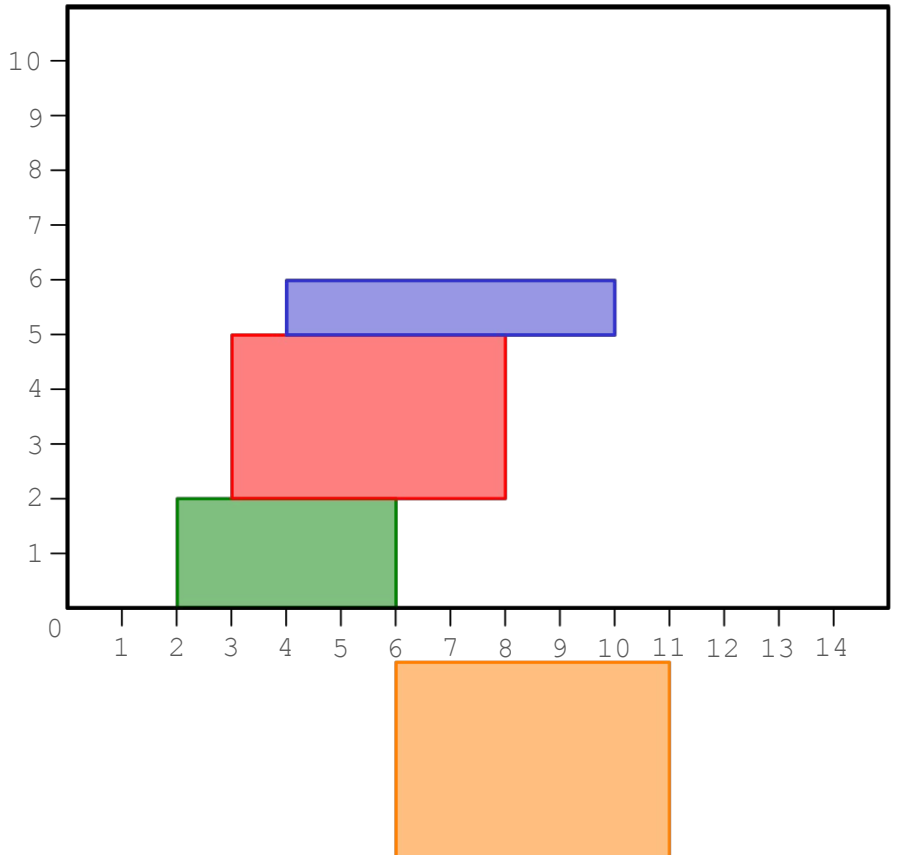
Exemplo

```
B1 = (2, 6, 2)
B2 = (3, 8, 3)
B3 = (4, 10, 1)
B4 = (6, 11, 4)
B5 = (9, 13, 3)
B6 = (11, 13, 1)
```



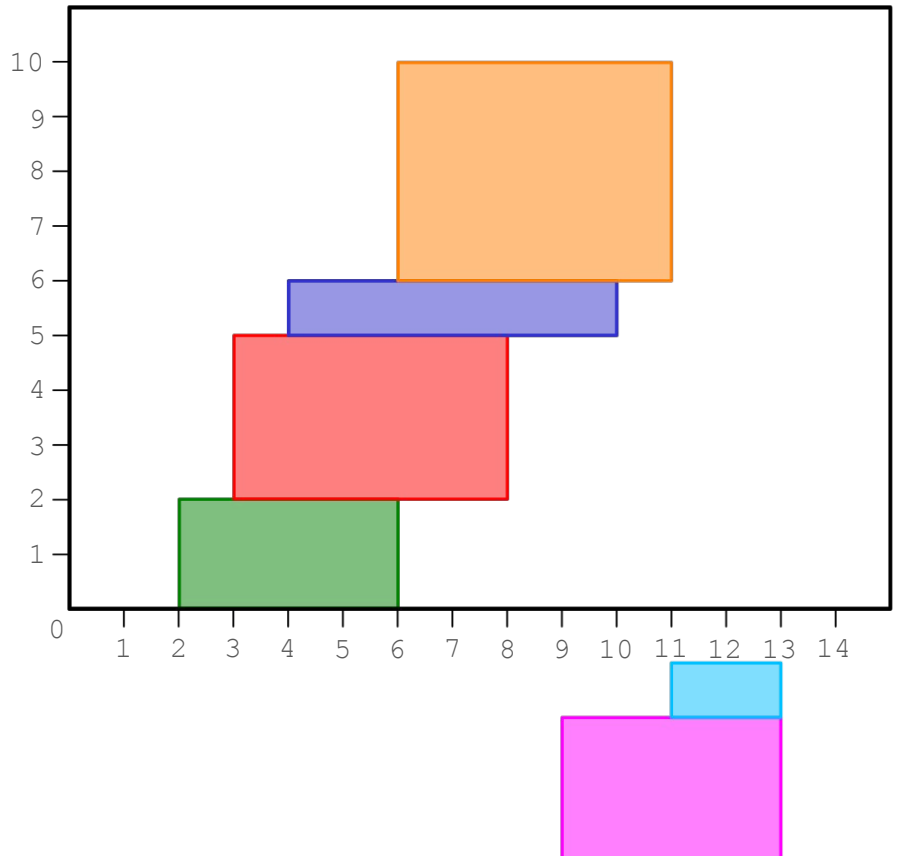
Exemplo

B1 = (2, 6, 2)
B2 = (3, 8, 3)
B3 = (4, 10, 1)
B4 = (6, 11, 4)
B5 = (9, 13, 3)
B6 = (11, 13, 1)



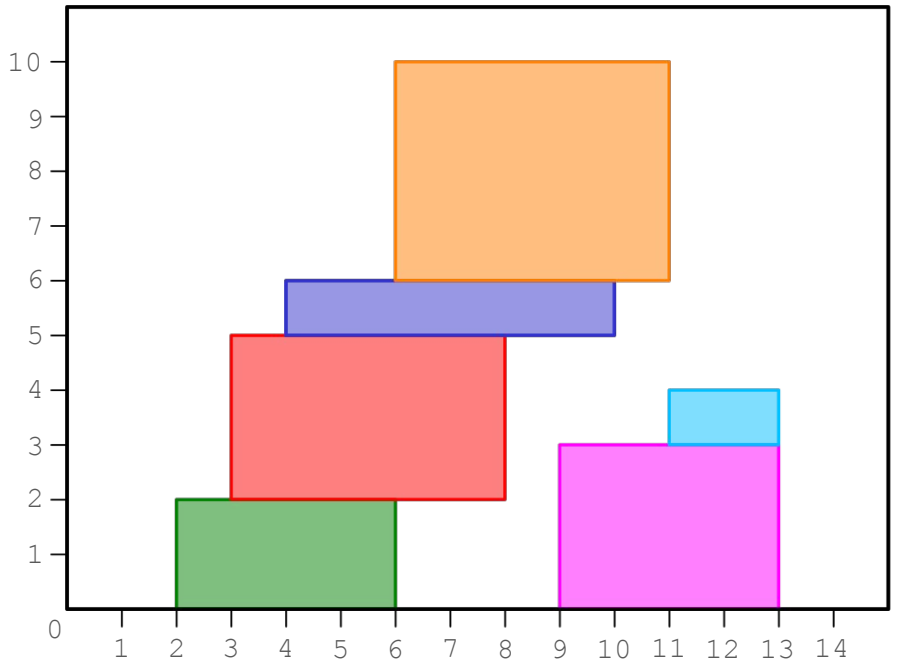
Exemplo

```
B1 = (2, 6, 2)
B2 = (3, 8, 3)
B3 = (4, 10, 1)
B4 = (6, 11, 4)
B5 = (9, 13, 3)
B6 = (11, 13, 1)
```



Exemplo

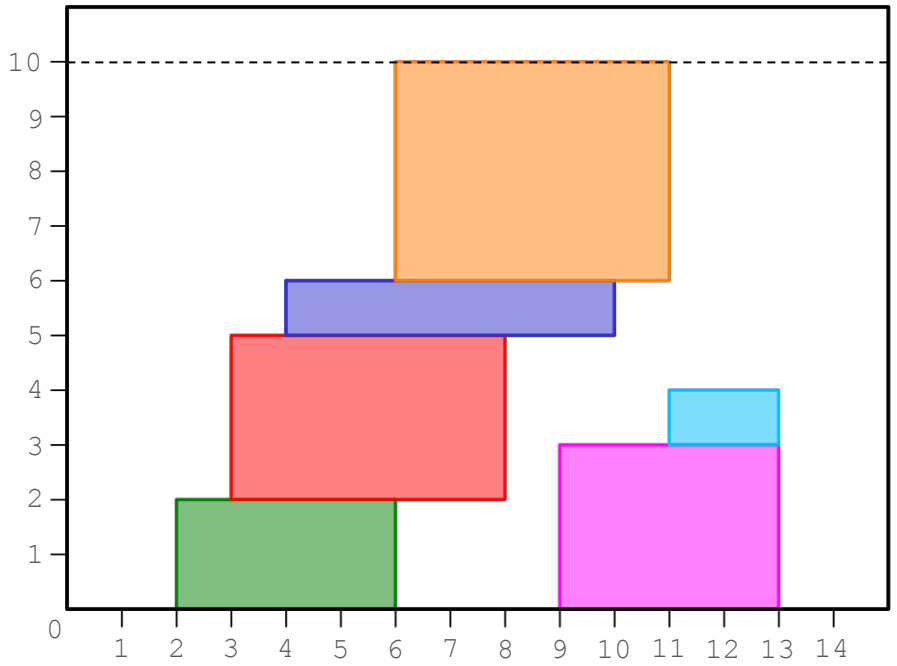
B1 = (2, 6, 2)
B2 = (3, 8, 3)
B3 = (4, 10, 1)
B4 = (6, 11, 4)
B5 = (9, 13, 3)
B6 = (11, 13, 1)



Exemplo

DCC

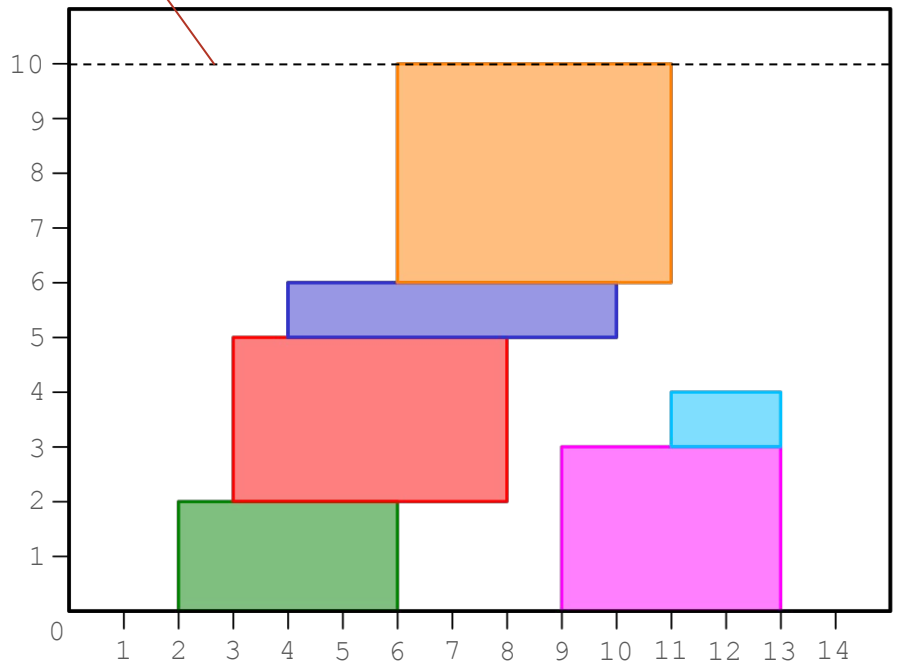
```
B1 = (2, 6, 2)
B2 = (3, 8, 3)
B3 = (4, 10, 1)
B4 = (6, 11, 4)
B5 = (9, 13, 3)
B6 = (11, 13, 1)
```



Exemplo

Tamanho da alocação: 10

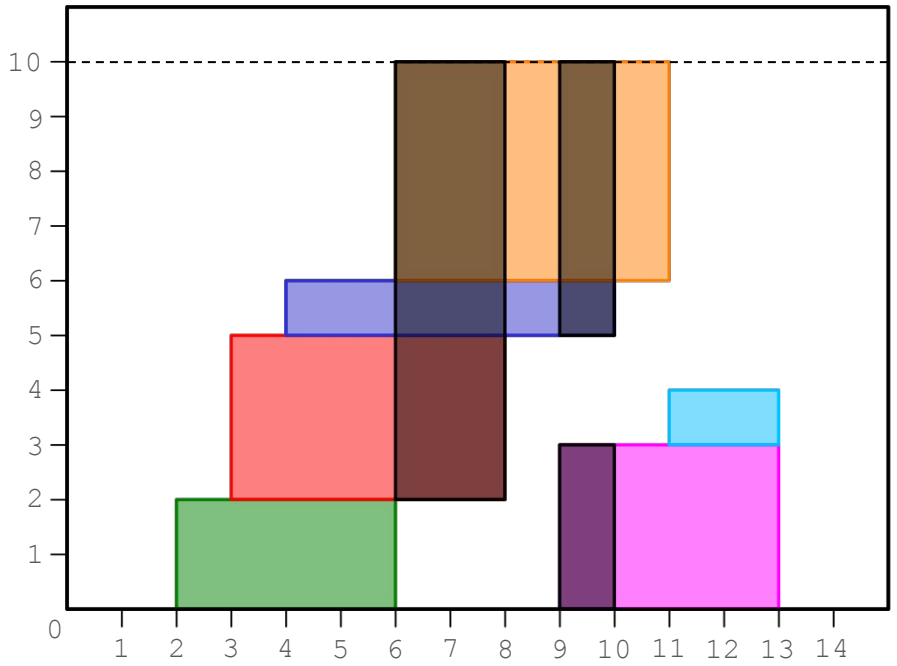
```
B1 = (2, 6, 2)
B2 = (3, 8, 3)
B3 = (4, 10, 1)
B4 = (6, 11, 4)
B5 = (9, 13, 3)
B6 = (11, 13, 1)
```



Exemplo

Tamanho da alocação: 10

B1 = (2, 6, 2)
B2 = (3, 8, 3)
B3 = (4, 10, 1)
B4 = (6, 11, 4)
B5 = (9, 13, 3)
B6 = (11, 13, 1)

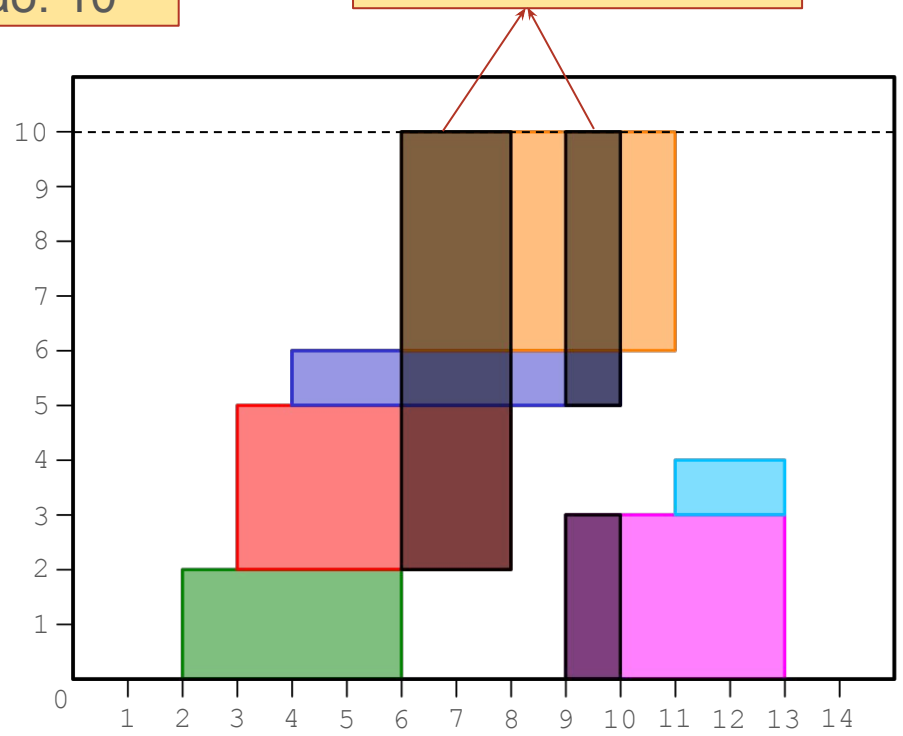


Exemplo

Tamanho da alocação: 10

Carga Máxima: 8

```
B1 = (2, 6, 2)
B2 = (3, 8, 3)
B3 = (4, 10, 1)
B4 = (6, 11, 4)
B5 = (9, 13, 3)
B6 = (11, 13, 1)
```



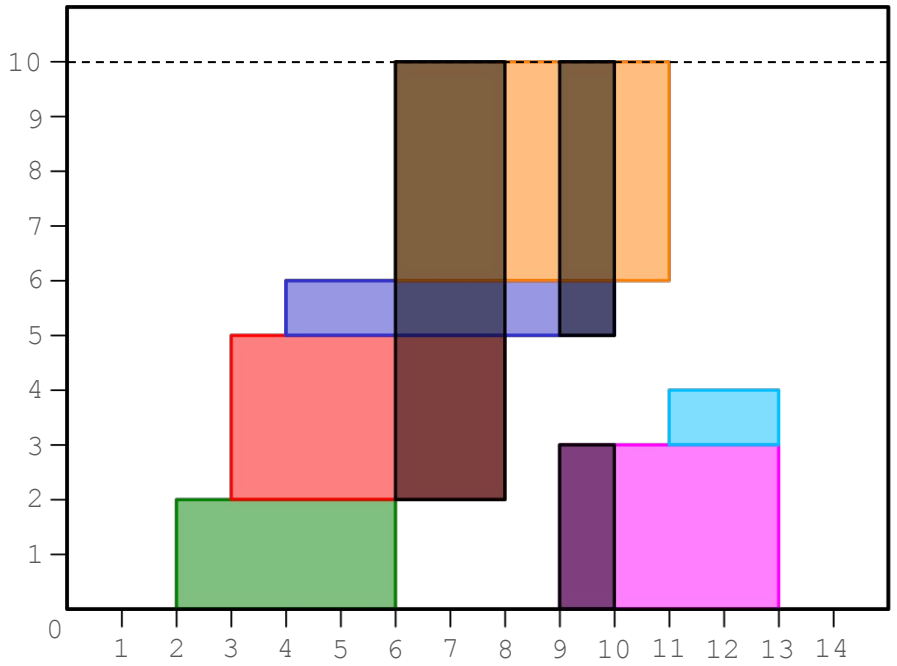
Exemplo

Tamanho da alocação: 10

Carga Máxima: 8

DCC

```
B1 = (2, 6, 2)
B2 = (3, 8, 3)
B3 = (4, 10, 1)
B4 = (6, 11, 4)
B5 = (9, 13, 3)
B6 = (11, 13, 1)
```



Fragmentação: 10 - 8 = 2

Problemas com o DSA

DCC

Problemas com o DSA

DCC

Existem algumas formas

Problemas com o DSA

DCC

Existem algumas formas na literatura e na indústria para resolver o DSA:

Problemas com o DSA

DCC

Existem algumas formas na literatura e na indústria para resolver o DSA:

- Minimalloc

Problemas com o DSA

DCC

Existem algumas formas na literatura e na indústria para resolver o DSA:

- Minimalloc
- Idealloc

Problemas com o DSA

DCC

Existem algumas formas na literatura e na indústria para resolver o DSA:

- Minimalloc
- Idealloc
- ApacheTVM

Problemas com o DSA

DCC

Existem algumas formas na literatura e na indústria para resolver o DSA:

- Minimalloc
- Idealloc
- ApacheTVM

Fragmentação :(

Problemas com o DSA

DCC

Existem algumas formas na literatura e na indústria para resolver o DSA:

- Minimalloc
- Idealloc
- ApacheTVM

Fragmentação :(

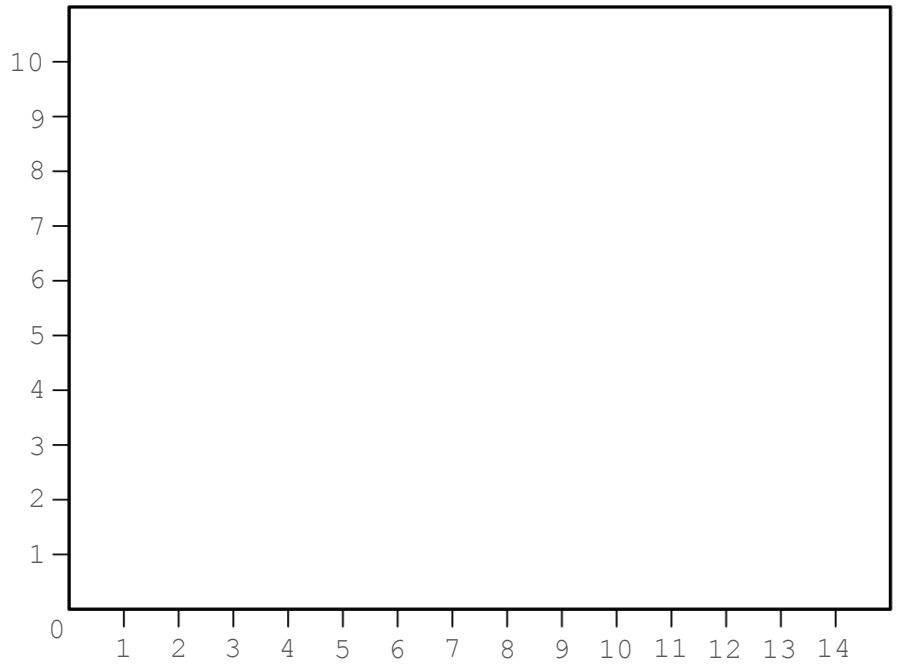
Vamos tentar defragmentar a memória

Exemplo - Defragmentação

DCC

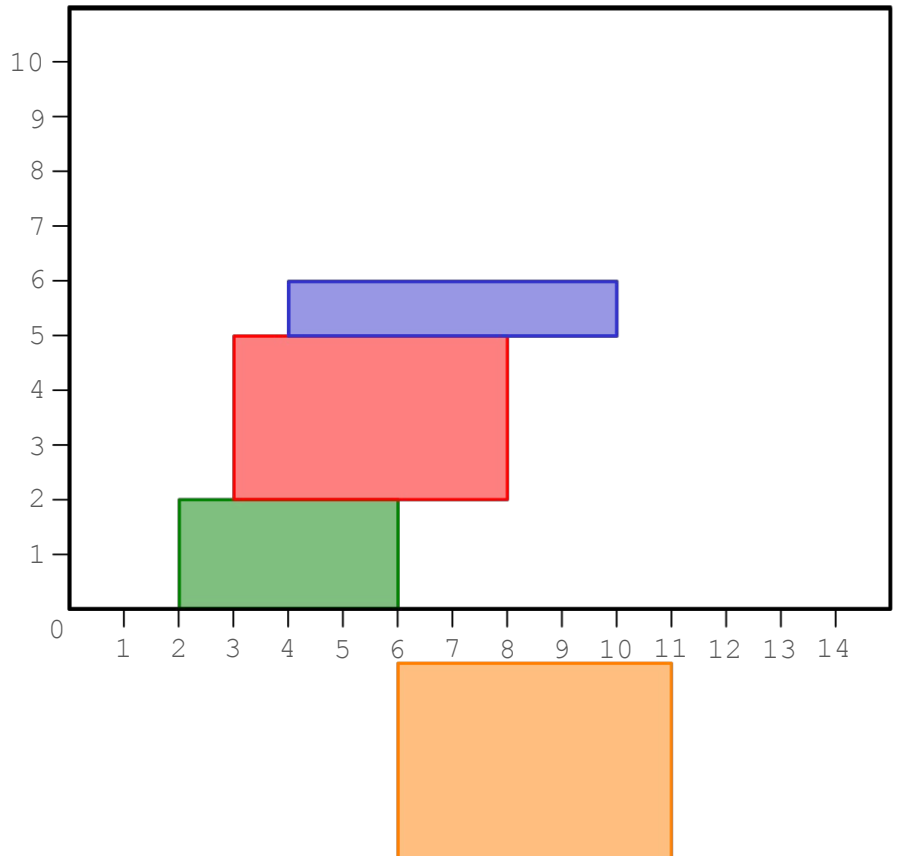
Exemplo - Defragmentação

```
B1  = (2, 6, 2)
B2  = (3, 8, 3)
B3  = (4, 10, 1)
B4  = (6, 11, 4)
B5  = (9, 13, 3)
B6  = (11, 13, 1)
```



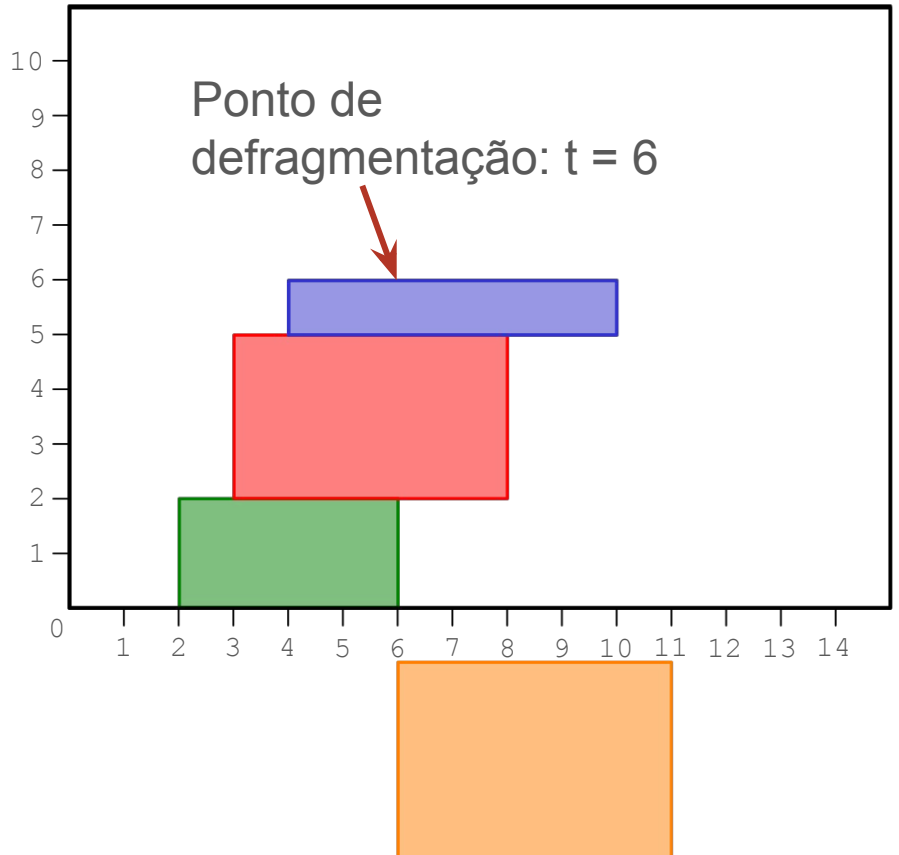
Exemplo - Defragmentação

```
B1  = (2, 6, 2)
B2  = (3, 8, 3)
B3  = (4, 10, 1)
B4  = (6, 11, 4)
B5  = (9, 13, 3)
B6  = (11, 13, 1)
```



Exemplo - Defragmentação

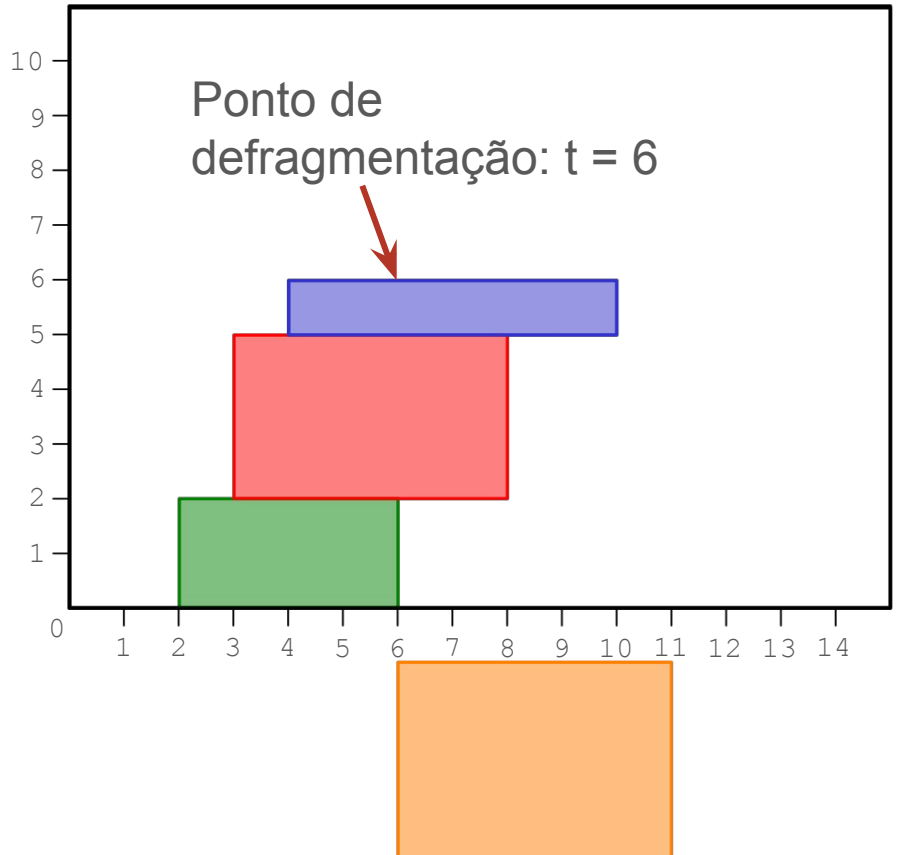
```
B1  = (2, 6, 2)
B2  = (3, 8, 3)
B3  = (4, 10, 1)
B4  = (6, 11, 4)
B5  = (9, 13, 3)
B6  = (11, 13, 1)
```



Exemplo - Defragmentação

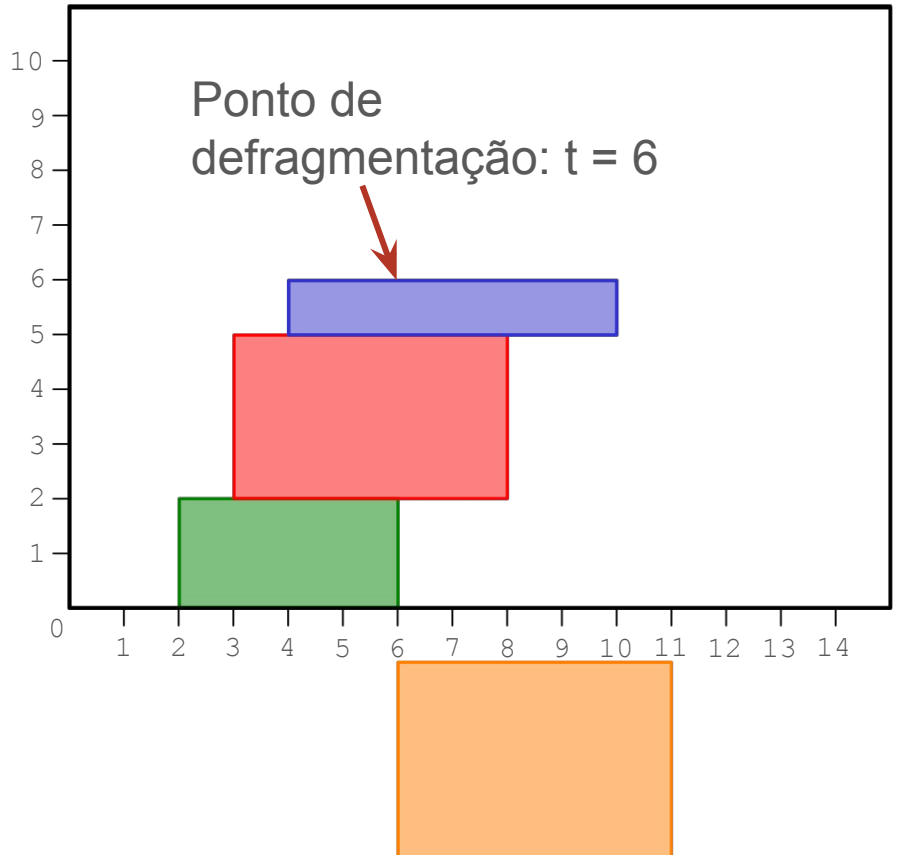
DCC

B1 = (2, 6, 2)
B2 = (3, 6, 3)
B2′ = (6, 8, 3)
B3 = (4, 10, 1)
B4 = (6, 11, 4)
B5 = (9, 13, 3)
B6 = (11, 13, 1)



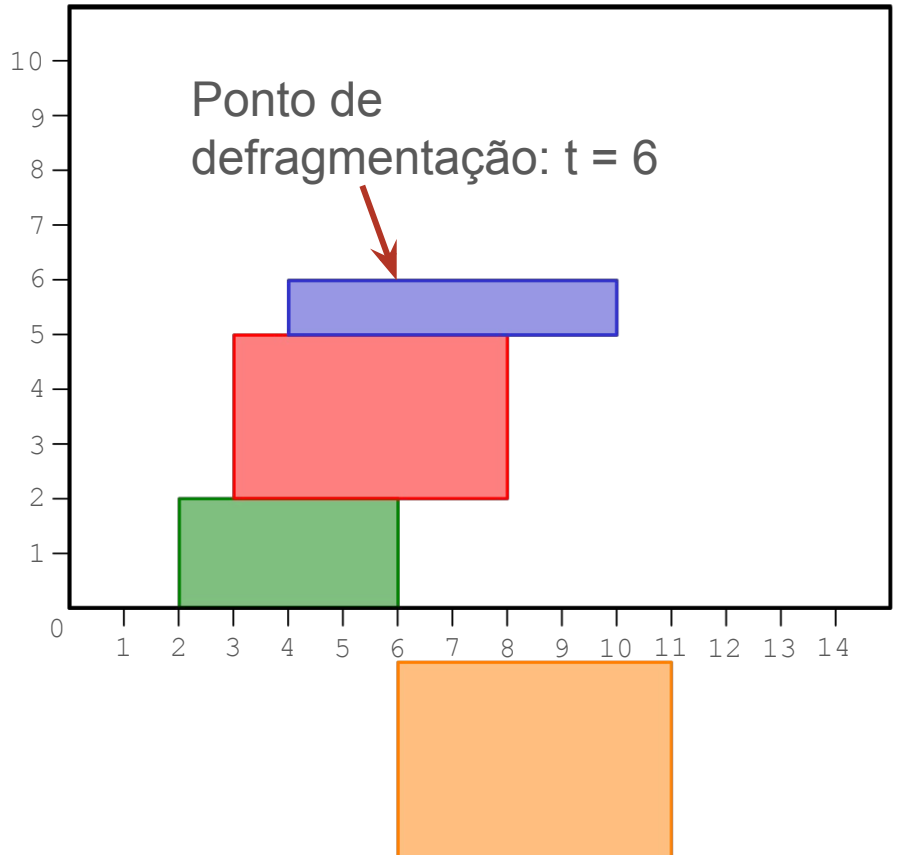
Exemplo - Defragmentação

B1 = (2, 6, 2)
B2 = (3, 6, 3)
B2′ = (6, 8, 3)
B3 = (4, 6, 1)
B3′ = (6, 10, 1)
B4 = (6, 11, 4)
B5 = (9, 13, 3)
B6 = (11, 13, 1)



Exemplo - Defragmentação

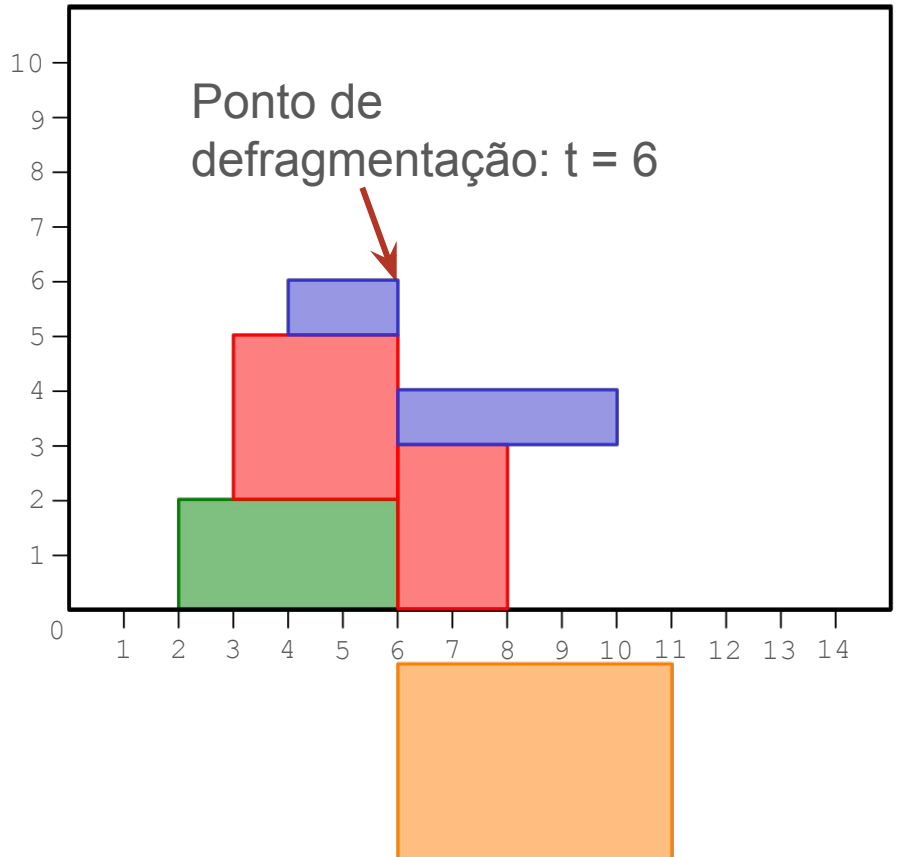
```
B1  = (2, 6, 2)
B2  = (3, 6, 3)
B2' = (6, 8, 3)
B3  = (4, 6, 1)
B3' = (6, 10, 1)
B4  = (6, 11, 4)
B5  = (9, 13, 3)
B6  = (11, 13, 1)
```



Exemplo - Defragmentação

DCC

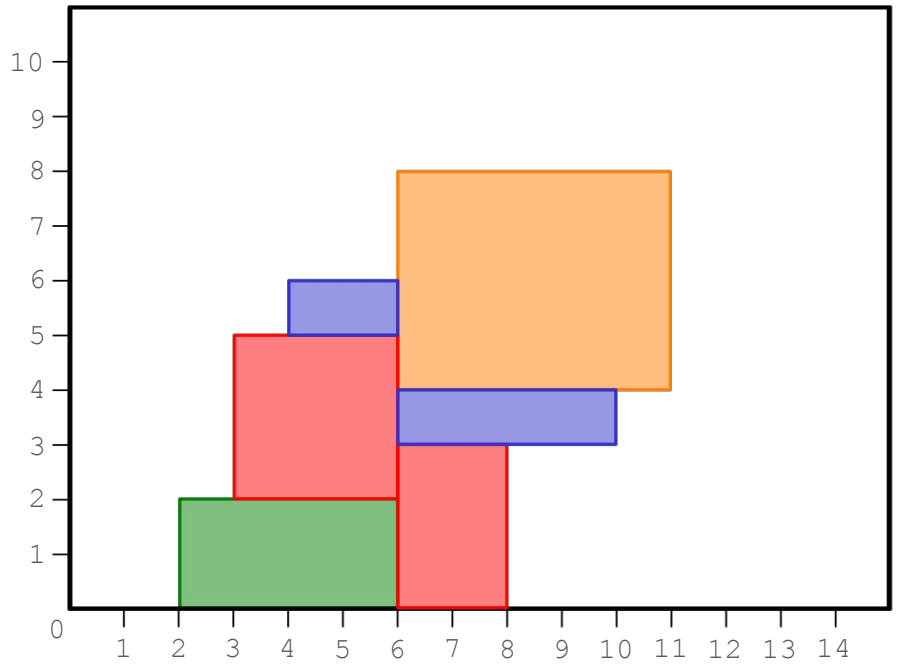
B1 = (2, 6, 2)
B2 = (3, 6, 3)
B2′ = (6, 8, 3)
B3 = (4, 6, 1)
B3′ = (6, 10, 1)
B4 = (6, 11, 4)
B5 = (9, 13, 3)
B6 = (11, 13, 1)



Exemplo - Defragmentação

DCC

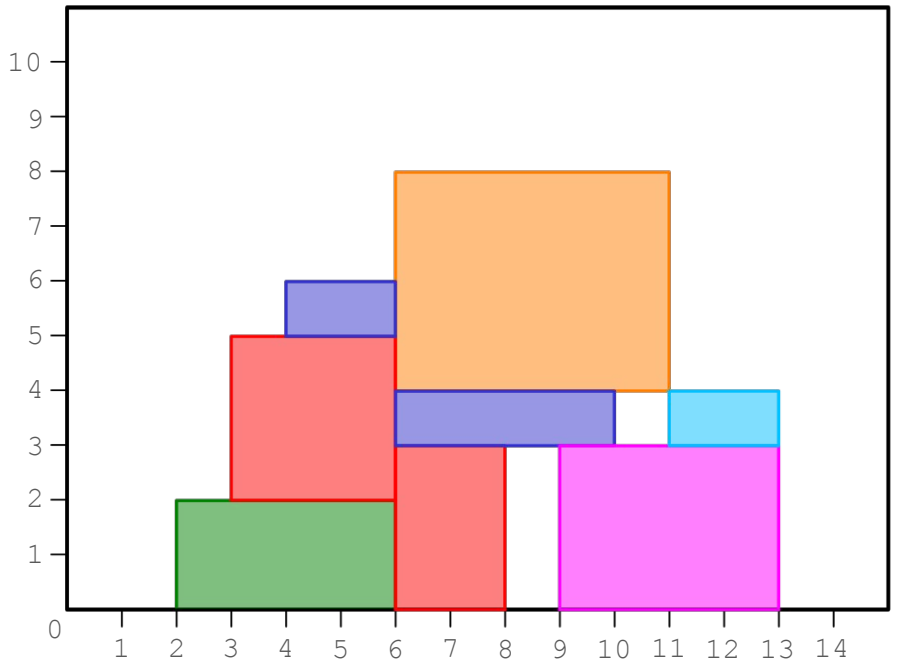
```
B1  = (2, 6, 2)
B2  = (3, 6, 3)
B2' = (6, 8, 3)
B3  = (4, 6, 1)
B3' = (6, 10, 1)
B4  = (6, 11, 4)
B5  = (9, 13, 3)
B6  = (11, 13, 1)
```



Exemplo - Defragmentação

DCC

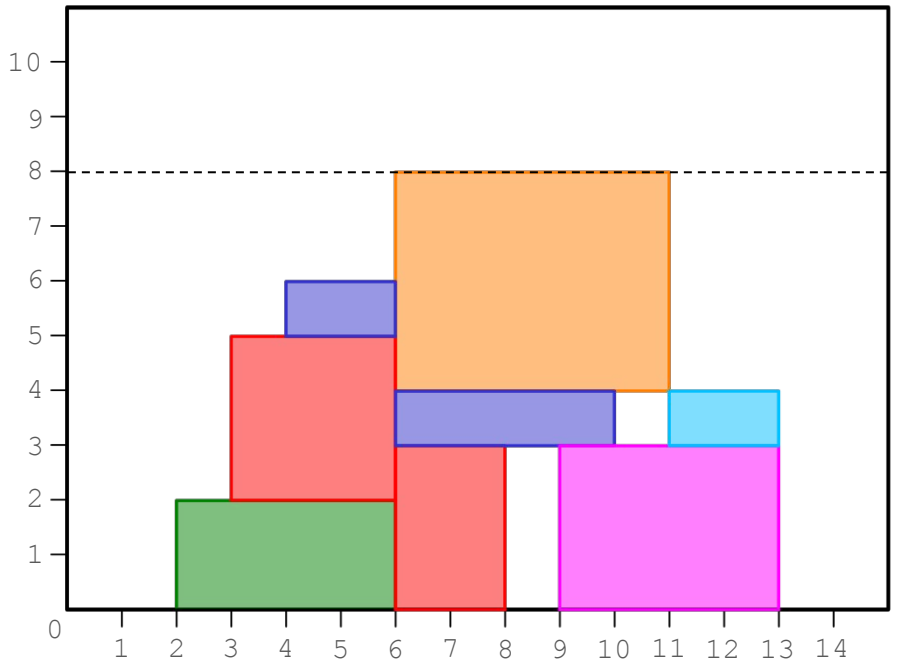
```
B1   = (2, 6, 2)
B2   = (3, 6, 3)
B2′  = (6, 8, 3)
B3   = (4, 6, 1)
B3′  = (6, 10, 1)
B4   = (6, 11, 4)
B5   = (9, 13, 3)
B6   = (11, 13, 1)
```



Exemplo - Defragmentação

DCC

```
B1   = (2, 6, 2)
B2   = (3, 6, 3)
B2′  = (6, 8, 3)
B3   = (4, 6, 1)
B3′  = (6, 10, 1)
B4   = (6, 11, 4)
B5   = (9, 13, 3)
B6   = (11, 13, 1)
```

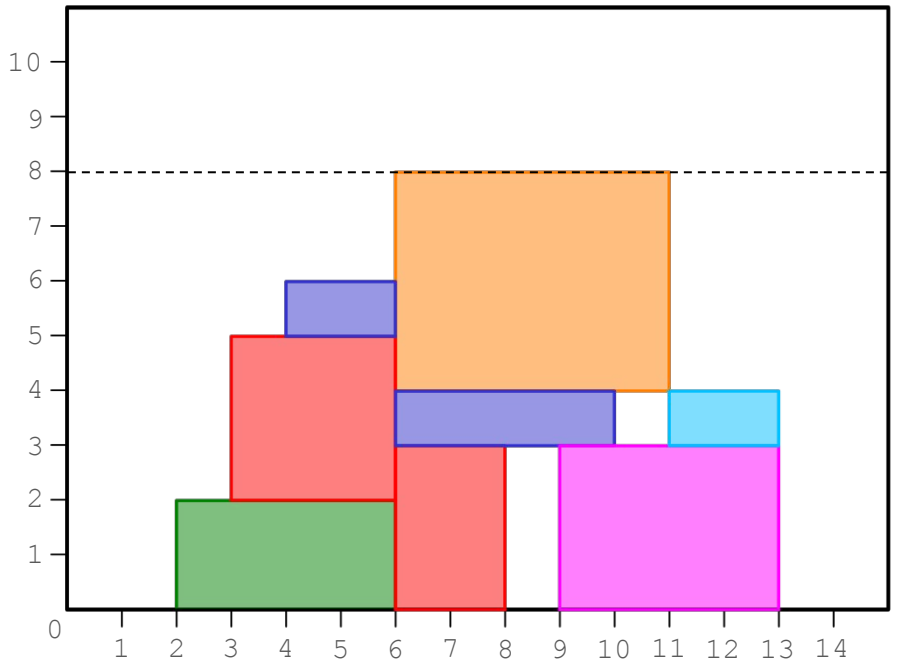


Exemplo - Defragmentação

Tamanho da alocação: 8

DCC

B1 = (2, 6, 2)
B2 = (3, 6, 3)
B2′ = (6, 8, 3)
B3 = (4, 6, 1)
B3′ = (6, 10, 1)
B4 = (6, 11, 4)
B5 = (9, 13, 3)
B6 = (11, 13, 1)



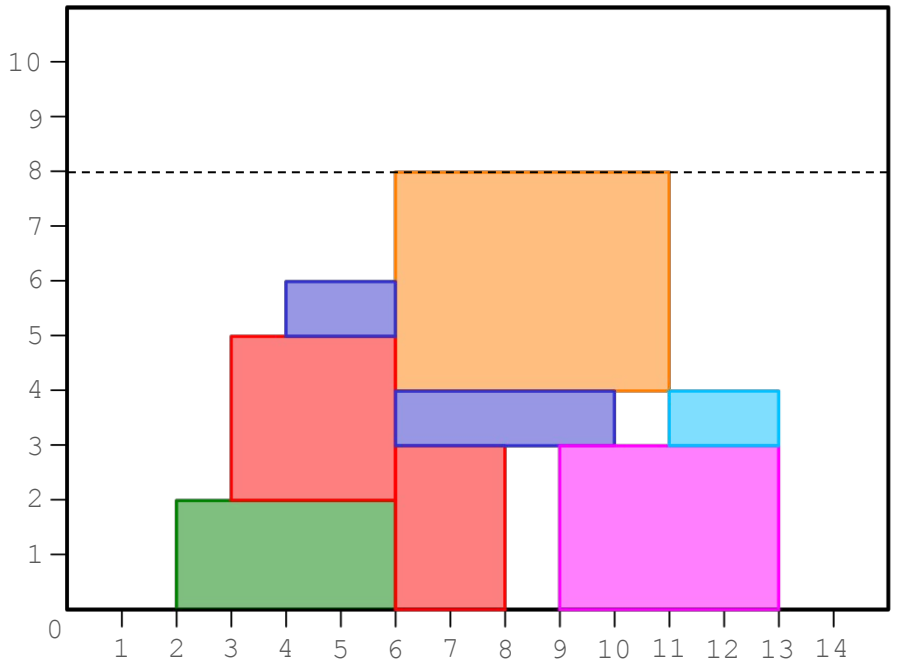
Exemplo - Defragmentação

Tamanho da alocação: 8

Carga Máxima: 8

DCC

```
B1   = (2, 6, 2)
B2   = (3, 6, 3)
B2′  = (6, 8, 3)
B3   = (4, 6, 1)
B3′  = (6, 10, 1)
B4   = (6, 11, 4)
B5   = (9, 13, 3)
B6   = (11, 13, 1)
```



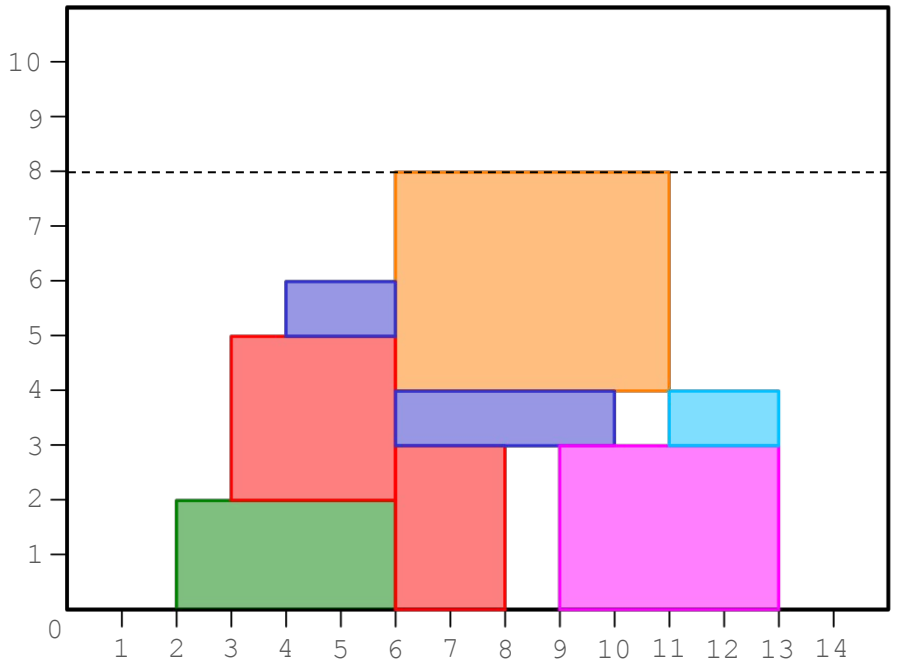
Exemplo - Defragmentação

Tamanho da alocação: 8

Carga Máxima: 8

DCC

```
B1   = (2, 6, 2)
B2   = (3, 6, 3)
B2′  = (6, 8, 3)
B3   = (4, 6, 1)
B3′  = (6, 10, 1)
B4   = (6, 11, 4)
B5   = (9, 13, 3)
B6   = (11, 13, 1)
```



Fragmentação:
8 - 8 = 0

Problemas Teóricos

DCC

Problemas Teóricos

DCC

1. Dada uma alocação M e um buffer B, qual é o menor número de movimentos de buffers alocados em M para alocar B em M?

Problemas Teóricos

DCC

1. Dada uma alocação M e um buffer B, qual é o menor número de movimentos de buffers alocados em M para alocar B em M?

Versão de decisão possivelmente NP-Completa (não provado ainda)

Problemas Teóricos

DCC

1. Dada uma alocação M e um buffer B, qual é o menor número de movimentos de buffers alocados em M para alocar B em M?
2. Dado um conjunto de buffers S e um limite de memória T, qual o menor número de defragmentações completas para fazer uma alocação M a partir de S com tamanho <= T?

Problemas Teóricos

DCC

1. Dada uma alocação M e um buffer B, qual é o menor número de movimentos de buffers alocados em M para alocar B em M?
2. Dado um conjunto de buffers S e um limite de memória T, qual o menor número de defragmentações completas para fazer uma alocação M a partir de S com tamanho <= T?

Versão de decisão NP-Completa

Problemas Teóricos

DCC

1. Dada uma alocação M e um buffer B, qual é o menor número de movimentos de buffers alocados em M para alocar B em M?
2. Dado um conjunto de buffers S e um limite de memória T, qual o menor número de defragmentações completas para fazer uma alocação M a partir de S com tamanho <= T?
3. Dado um conjunto de buffers S e um limite de memória T, qual o menor número de movimentos de buffers para fazer uma alocação M a partir de S com tamanho <= T?

Problemas Teóricos

DCC

1. Dada uma alocação M e um buffer B, qual é o menor número de movimentos de buffers alocados em M para alocar B em M?
2. Dado um conjunto de buffers S e um limite de memória T, qual o menor número de defragmentações completas para fazer uma alocação M a partir de S com tamanho <= T?
3. Dado um conjunto de buffers S e um limite de memória T, qual o menor número de movimentos de buffers para fazer uma alocação M a partir de S com tamanho <= T?

Problemas Teóricos

DCC

1. Dada uma alocação M e um buffer B, qual é o menor número de movimentos de buffers alocados em M para alocar B em M?
2. Dado um conjunto de buffers S e um limite de memória T, qual o menor número de defragmentações completas para fazer uma alocação M a partir de S com tamanho <= T?
3. Dado um conjunto de buffers S e um limite de memória T, qual o menor número de movimentos de buffers para fazer uma alocação M a partir de S com tamanho <= T?

Versão de decisão NP-Completa

Alguns avanços

DCC

Alguns avanços

O trabalho ainda não possui resultados,

Alguns avanços

DCC

O trabalho ainda não possui resultados, porém algumas contribuições

Alguns avanços

O trabalho ainda não possui resultados, porém algumas contribuições já foram feitas:

Alguns avanços

DCC

O trabalho ainda não possui resultados, porém algumas contribuições já foram feitas:

- A prova de NP-completude dos problemas teóricos;

Alguns avanços

DCC

O trabalho ainda não possui resultados, porém algumas contribuições já foram feitas:

- A prova de NP-completude dos problemas teóricos;
- Implementação em MLIR; e



Alguns avanços

DCC

O trabalho ainda não possui resultados, porém algumas contribuições já foram feitas:

- A prova de NP-completude dos problemas teóricos;
- Implementação em MLIR; e
- Implementação em C++ do idealloc





Próximos passos

Próximos passos

DCC

Implementação e comparação de heurísticas de defragmentação.

Contato

Elisa Fröhlich

elisa.frohlich@dcc.ufmg.br

https://homepages.dcc.ufmg.br/~elisa.frohlich/



Project Raróg

A Static Heap Allocator for Bounded Neural Networks

UFMG DCC DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO Compilers Laboratory