

ALGORITMOS E ESTRUTURAS DE DADOS II

Ordenação Externa
Karina Valdivia Delgado

Ordenação externa

- Processo de ordenação de arquivos muito grandes, cujo índice não cabe na memória principal. Faremos a ordenação com auxílio de algum dispositivo de memória externa.
- O objetivo é ordenar os registros com o menor número de acessos ao disco (operações de leitura ou escrita em disco)

Algoritmo de intercalação balanceada

- **Ordena os registros usando intercalação de N caminhos (vias/fontes) balanceados.**
- **A quantidade de fontes está relacionada com a quantidade de HDs externos que tenho disponível.**
- **No caso de usar apenas um HD, a quantidade de fontes está relacionada com a quantidade de arquivos que serão usados.**

Algoritmo de intercalação balanceada

- Fases:
 - Distribuição de **blocos de registros ordenados (corridas)** por N caminhos balanceados
 - Intercalação das corridas geradas dos N caminhos até que uma corrida seja obtida

Algoritmo de intercalação balanceada

6	12	15	11	3	7	29	35	42	55	45	65	76	89	8	4	22	24	23	45	89	99	88	76	48	78	32	1	2	20	30	16
---	----	----	----	---	---	----	----	----	----	----	----	----	----	---	---	----	----	----	----	----	----	----	----	----	----	----	---	---	----	----	----

Fase 1: Distribuição de **blocos de registros ordenados (corridas)** por N caminhos balanceados

- caminhos(fontes) 2 ou seja vamos usar 2 conjuntos de 2 arquivos temporários
- blocos ordenados (corridas) de tamanho 4, ou seja, estamos fazendo a suposição que temos capacidade para ordenar em memória principal apenas 4 registros.

Arquivo 1

6	11	12	15
---	----	----	----

42	45	55	65
----	----	----	----

22	23	24	45
----	----	----	----

1	32	48	78
---	----	----	----

Arquivo 2

3	7	29	35
---	---	----	----

8	4	76	89
---	---	----	----

76	88	89	99
----	----	----	----

2	16	20	30
---	----	----	----

Algoritmo de intercalação balanceada

Fase 2: Intercalação das corridas geradas dos N caminhos até que uma corrida seja obtida

Arquivo 1

6	11	12	15
---	----	----	----

42	45	55	65
----	----	----	----

22	23	24	45
----	----	----	----

1	32	48	78
---	----	----	----

Arquivo 2

3	7	29	35
---	---	----	----

8	4	76	89
---	---	----	----

76	88	89	99
----	----	----	----

2	16	20	30
---	----	----	----

Primeira intercalação: são gerados blocos de tamanho 8

Arquivo 3

3	6	7	11	12	15	29	35
---	---	---	----	----	----	----	----

22	23	24	45	76	88	89	99
----	----	----	----	----	----	----	----

Arquivo 4

4	8	42	45	55	65	76	89
---	---	----	----	----	----	----	----

1	2	16	20	30	32	48	78
---	---	----	----	----	----	----	----

Obs: Não existe um separador físico entre as corridas, mas como sabemos qual o tamanho de cada bloco, sabemos quando o bloco acaba.

Algoritmo de intercalação balanceada

Fase 2: Intercalação das corridas geradas dos N caminhos até que uma corrida seja obtida

Arquivo 3

3	6	7	11	12	15	29	35
---	---	---	----	----	----	----	----

22	23	24	45	76	88	89	99
----	----	----	----	----	----	----	----

Arquivo 4

4	8	42	45	55	65	76	89
---	---	----	----	----	----	----	----

1	2	16	20	30	32	48	78
---	---	----	----	----	----	----	----

Segunda intercalação: são gerados blocos de tamanho 16

Arquivo 1

3	4	6	7	8	11	12	15	29	35	42	45	55	65	76	89
---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----

Arquivo 2

1	2	16	20	22	23	24	30	32	45	48	76	78	88	89	99
---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Algoritmo de intercalação balanceada

Fase 2: Intercalação das corridas geradas dos N caminhos até que uma corrida seja obtida

Arquivo 1

3	4	6	7	8	11	12	15	29	35	42	45	55	65	76	89
---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----

Arquivo 2

1	2	16	20	22	23	24	30	32	45	48	76	78	88	89	99
---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Terceira intercalação: são gerados blocos de tamanho 32

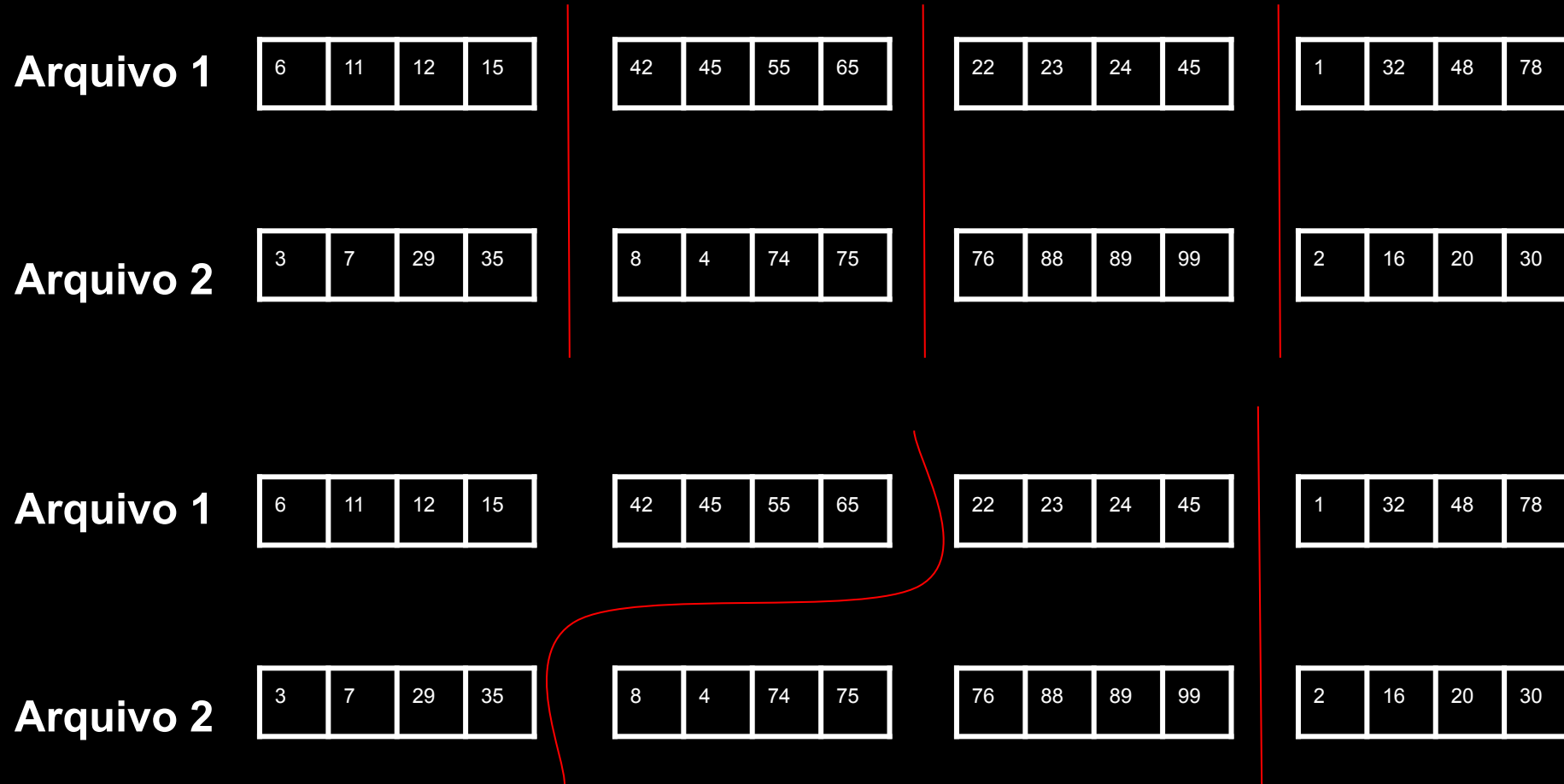
Arquivo 3

1	2	3	4	6	7	8	11	12	15	16	20	22	23	24	29	30	32	35	42	45	45	48	55	65	76	76	78	88	89	89	99
---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Arquivo 4

Segmentos de tamanho variável

Primeira estratégia de otimização: aproveitar a eventual ordenação entre os blocos

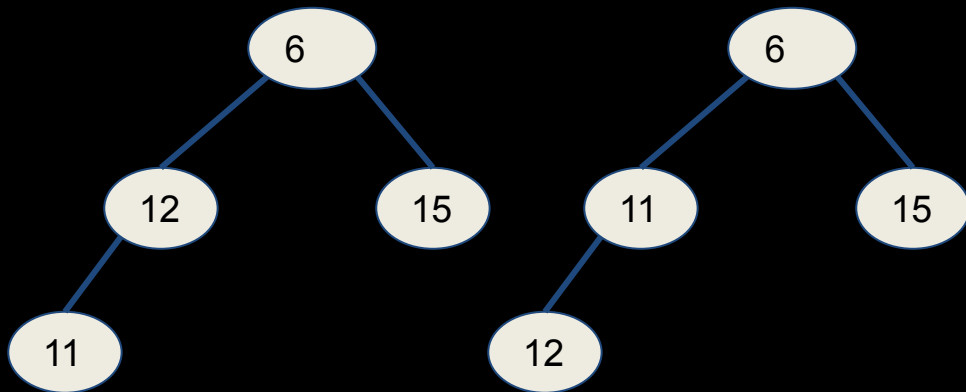


Intercalação com seleção por substituição

- Segunda estratégia de otimização: gerar segmentos ordenados maiores na fase de distribuição
- Para isso, será utilizada uma fila de prioridades representada como um heap

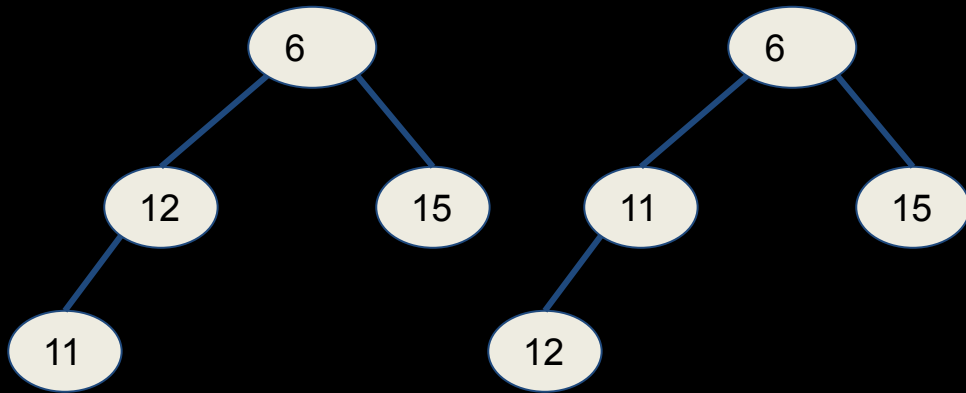
Intercalação com seleção por substituição

6	12	15	11	3	7	29	35	42	55	45	65	76	89	8	4	22	24	23	45	89	99	88	76	48	78	32	1	2	20	30	16
---	----	----	----	---	---	----	----	----	----	----	----	----	----	---	---	----	----	----	----	----	----	----	----	----	----	----	---	---	----	----	----



Intercalação com seleção por substituição

6	12	15	11	3	7	29	35	42	55	45	65	76	89	8	4	22	24	23	45	89	99	88	76	48	78	32	1	2	20	30	16
---	----	----	----	---	---	----	----	----	----	----	----	----	----	---	---	----	----	----	----	----	----	----	----	----	----	----	---	---	----	----	----

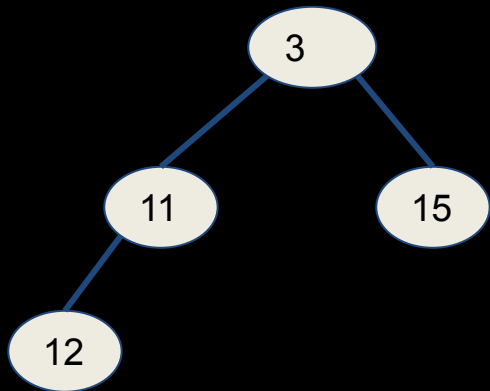


Arq 1: 6

Arq 2:

Intercalação com seleção por substituição

6	12	15	11	3	7	29	35	42	55	45	65	76	89	8	4	22	24	23	45	89	99	88	76	48	78	32	1	2	20	30	16
---	----	----	----	---	---	----	----	----	----	----	----	----	----	---	---	----	----	----	----	----	----	----	----	----	----	----	---	---	----	----	----

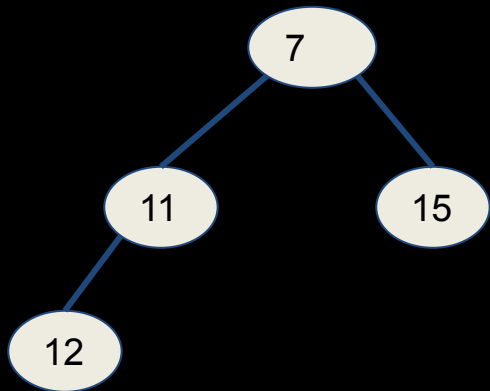


Arq 1: 6 3

Arq 2:

Intercalação com seleção por substituição

6	12	15	11	3	7	29	35	42	55	45	65	76	89	8	4	22	24	23	45	89	99	88	76	48	78	32	1	2	20	30	16
---	----	----	----	---	---	----	----	----	----	----	----	----	----	---	---	----	----	----	----	----	----	----	----	----	----	----	---	---	----	----	----

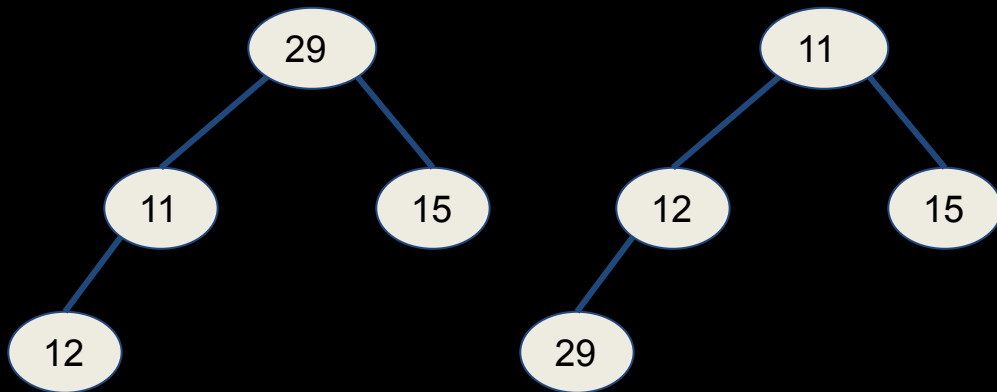


Arq 1: 6 3 7

Arq 2:

Intercalação com seleção por substituição

6	12	15	11	3	7	29	35	42	55	45	65	76	89	8	4	22	24	23	45	89	99	88	76	48	78	32	1	2	20	30	16
---	----	----	----	---	---	----	----	----	----	----	----	----	----	---	---	----	----	----	----	----	----	----	----	----	----	----	---	---	----	----	----

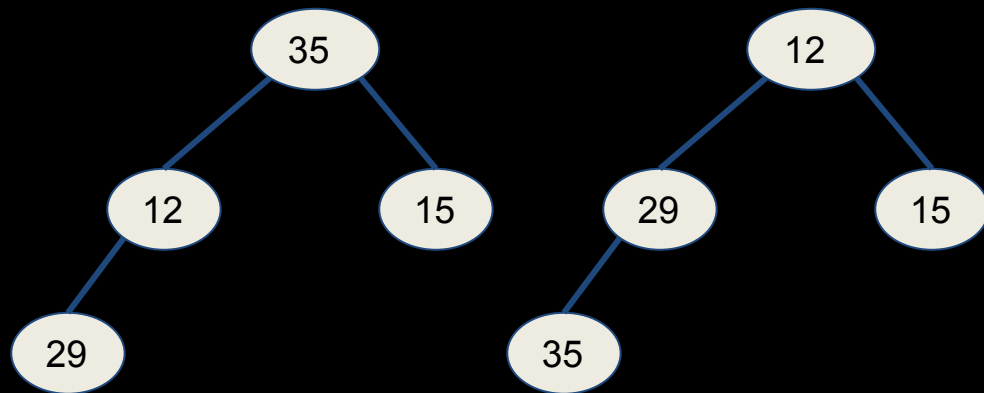


Arq 1: 6 3 7 11

Arq 2:

Intercalação com seleção por substituição

6	12	15	11	3	7	29	35	42	55	45	65	76	89	8	4	22	24	23	45	89	99	88	76	48	78	32	1	2	20	30	16
---	----	----	----	---	---	----	----	----	----	----	----	----	----	---	---	----	----	----	----	----	----	----	----	----	----	----	---	---	----	----	----

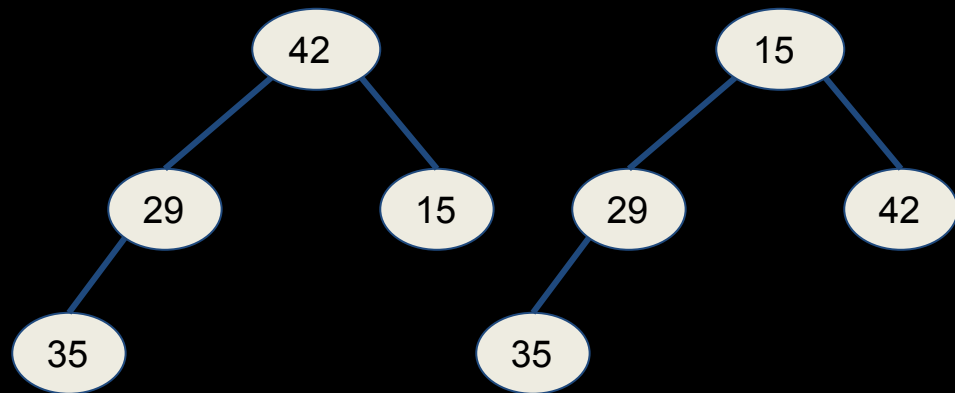


Arq 1: 6 3 7 11 12

Arq 2:

Intercalação com seleção por substituição

6	12	15	11	3	7	29	35	42	55	45	65	76	89	8	4	22	24	23	45	89	99	88	76	48	78	32	1	2	20	30	16
---	----	----	----	---	---	----	----	----	----	----	----	----	----	---	---	----	----	----	----	----	----	----	----	----	----	----	---	---	----	----	----

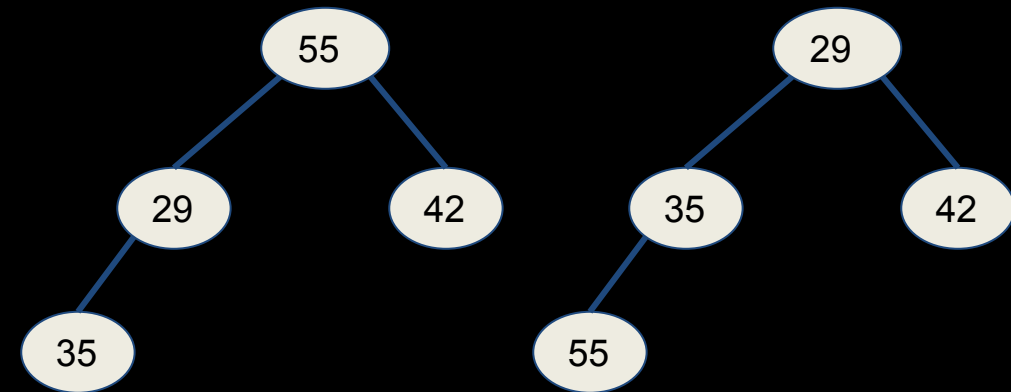


Arq 1: 6 3 7 11 12 15

Arq 2:

Intercalação com seleção por substituição

6	12	15	11	3	7	29	35	42	55	45	65	76	89	8	4	22	24	23	45	89	99	88	76	48	78	32	1	2	20	30	16
---	----	----	----	---	---	----	----	----	----	----	----	----	----	---	---	----	----	----	----	----	----	----	----	----	----	----	---	---	----	----	----

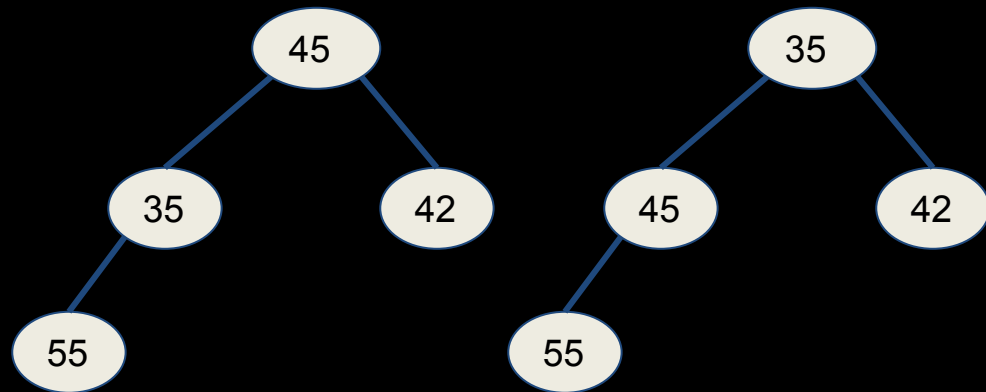


Arq 1: 6 3 7 11 12 15 29

Arq 2:

Intercalação com seleção por substituição

6	12	15	11	3	7	29	35	42	55	45	65	76	89	8	4	22	24	23	45	89	99	88	76	48	78	32	1	2	20	30	16
---	----	----	----	---	---	----	----	----	----	----	----	----	----	---	---	----	----	----	----	----	----	----	----	----	----	----	---	---	----	----	----

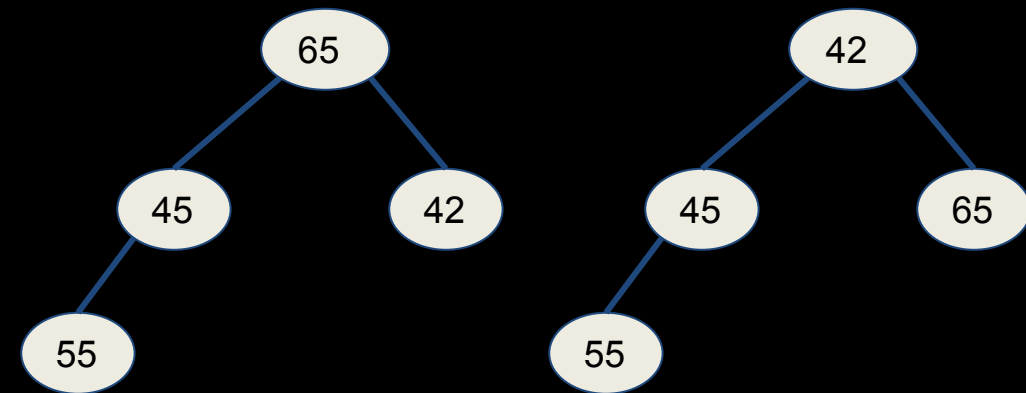


Arq 1: 6 3 7 11 12 15 29 35

Arq 2:

Intercalação com seleção por substituição

6	12	15	11	3	7	29	35	42	55	45	65	76	89	8	4	22	24	23	45	89	99	88	76	48	78	32	1	2	20	30	16
---	----	----	----	---	---	----	----	----	----	----	----	----	----	---	---	----	----	----	----	----	----	----	----	----	----	----	---	---	----	----	----

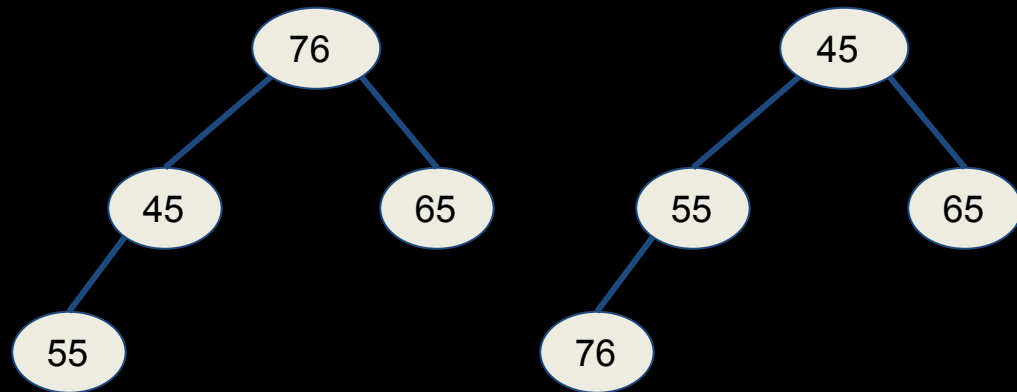


Arq 1: 6 3 7 11 12 15 29 35 42

Arq 2:

Intercalação com seleção por substituição

6	12	15	11	3	7	29	35	42	55	45	65	76	89	8	4	22	24	23	45	89	99	88	76	48	78	32	1	2	20	30	16
---	----	----	----	---	---	----	----	----	----	----	----	----	----	---	---	----	----	----	----	----	----	----	----	----	----	----	---	---	----	----	----

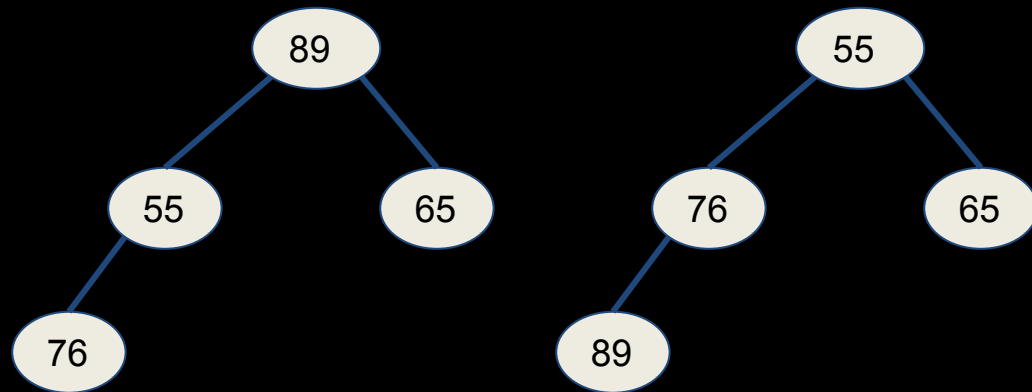


Arq 1: 6 3 7 11 12 15 29 35 42 45

Arq 2:

Intercalação com seleção por substituição

6	12	15	11	3	7	29	35	42	55	45	65	76	89	8	4	22	24	23	45	89	99	88	76	48	78	32	1	2	20	30	16
---	----	----	----	---	---	----	----	----	----	----	----	----	----	---	---	----	----	----	----	----	----	----	----	----	----	----	---	---	----	----	----

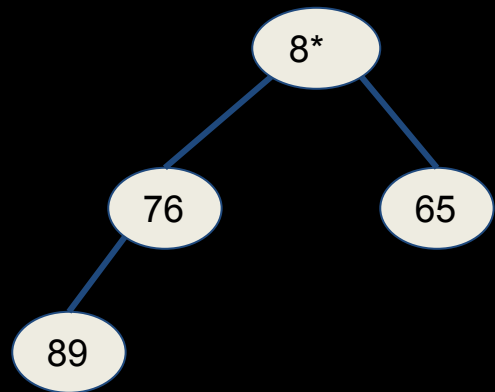


Arq 1: 6 3 7 11 12 15 29 35 42 45 55

Arq 2:

Intercalação com seleção por substituição

6	12	15	11	3	7	29	35	42	55	45	65	76	89	8	4	22	24	23	45	89	99	88	76	48	78	32	1	2	20	30	16
---	----	----	----	---	---	----	----	----	----	----	----	----	----	---	---	----	----	----	----	----	----	----	----	----	----	----	---	---	----	----	----



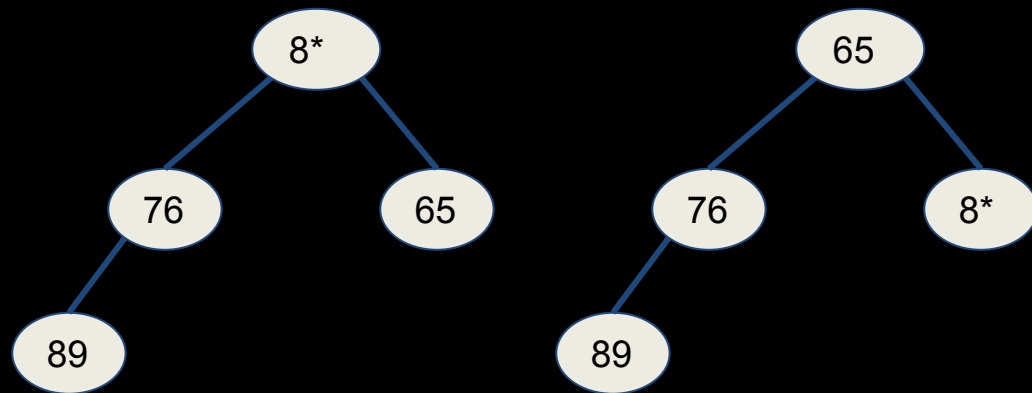
Arq 1: 6 3 7 11 12 15 29 35 42 45 55

Arq 2:

$8 < 55$, se eu insiro 8 no arquivo 1, perderia a ordenação dos meus elementos e o bloco teria que terminar aqui em 55. Porém, existem ainda outros elementos que poderiam entrar no meu bloco e que estão no heap (65,76,89). Então vamos colocar uma marcação de novo bloco para 8, será 8*

Intercalação com seleção por substituição

6	12	15	11	3	7	29	35	42	55	45	65	76	89	8	4	22	24	23	45	89	99	88	76	48	78	32	1	2	20	30	16
---	----	----	----	---	---	----	----	----	----	----	----	----	----	---	---	----	----	----	----	----	----	----	----	----	----	----	---	---	----	----	----



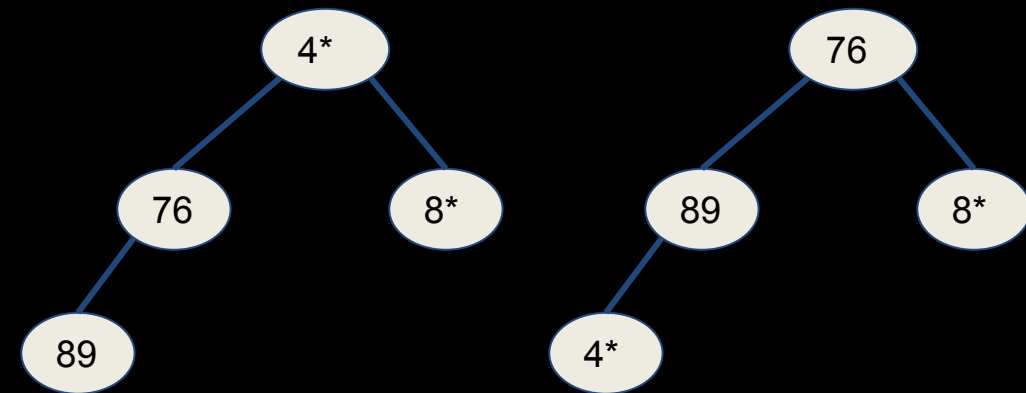
Arq 1: 6 3 7 11 12 15 29 35 42 45 55 65

Arq 2:

8 < 55, se eu insiro 8 no arquivo 1, perderia a ordenação dos meus elementos e o bloco teria que terminar aqui em 55. Porém, existem ainda outros elementos que poderiam entrar no meu bloco e que estão no heap (65,76,89). Então vamos colocar uma marcação de novo bloco para 8, será 8*

Intercalação com seleção por substituição

6	12	15	11	3	7	29	35	42	55	45	65	76	89	8	4	22	24	23	45	89	99	88	76	48	78	32	1	2	20	30	16
---	----	----	----	---	---	----	----	----	----	----	----	----	----	---	---	----	----	----	----	----	----	----	----	----	----	----	---	---	----	----	----



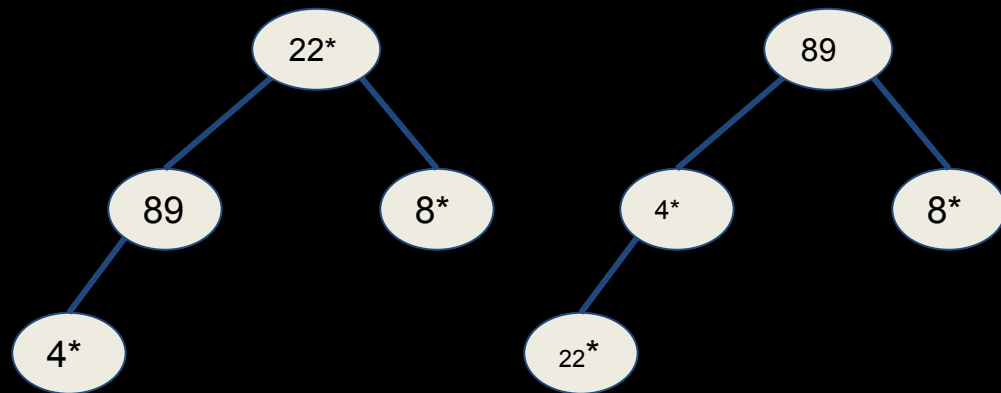
Arq 1: 6 3 7 11 12 15 29 35 42 45 55 65 76

Arq 2:

$4 < 65$, então vamos colocar uma marcação de novo bloco para 4, será 4*

Intercalação com seleção por substituição

6	12	15	11	3	7	29	35	42	55	45	65	76	89	8	4	22	24	23	45	89	99	88	76	48	78	32	1	2	20	30	16
---	----	----	----	---	---	----	----	----	----	----	----	----	----	---	---	----	----	----	----	----	----	----	----	----	----	----	---	---	----	----	----



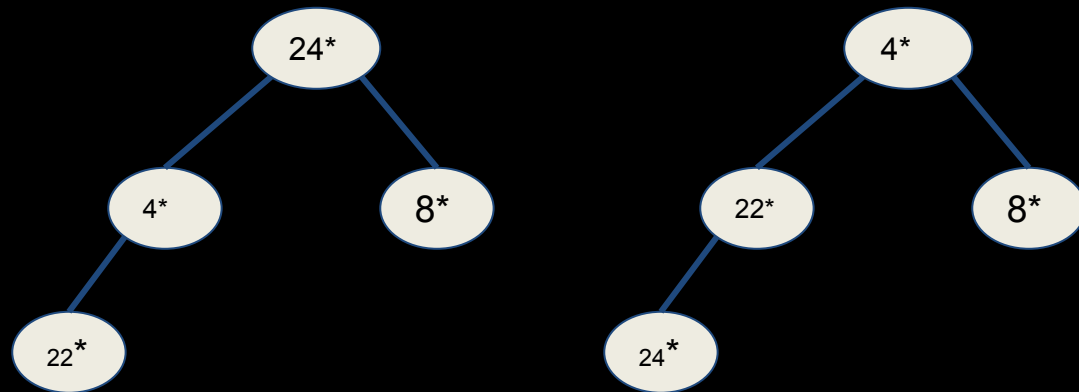
Arq 1: 6 3 7 11 12 15 29 35 42 45 55 65 76 89

Arq 2:

$22 < 76$, então vamos colocar uma marcação de novo bloco para 22, será 22*

Intercalação com seleção por substituição

6	12	15	11	3	7	29	35	42	55	45	65	76	89	8	4	22	24	23	45	89	99	88	76	48	78	32	1	2	20	30	16
---	----	----	----	---	---	----	----	----	----	----	----	----	----	---	---	----	----	----	----	----	----	----	----	----	----	----	---	---	----	----	----



Arq 1: 6 3 7 11 12 15 29 35 42 45 55 65 76 89

Arq 2: 4*

$24 < 89$, então vamos colocar uma marcação de novo bloco para 24, será 24*

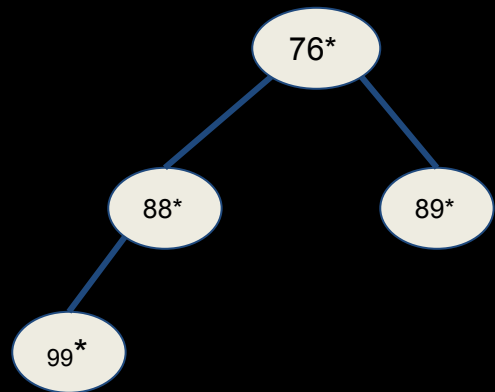
Note que todas as chaves no heap são do bloco * => ao retirar 4* ele deverá ir para o arquivo 2

Intercalação com seleção por substituição

Continua ...

Intercalação com seleção por substituição

6	12	15	11	3	7	29	35	42	55	45	65	76	89	8	4	22	24	23	45	89	99	88	76	48	78	32	1	2	20	30	16
---	----	----	----	---	---	----	----	----	----	----	----	----	----	---	---	----	----	----	----	----	----	----	----	----	----	----	---	---	----	----	----

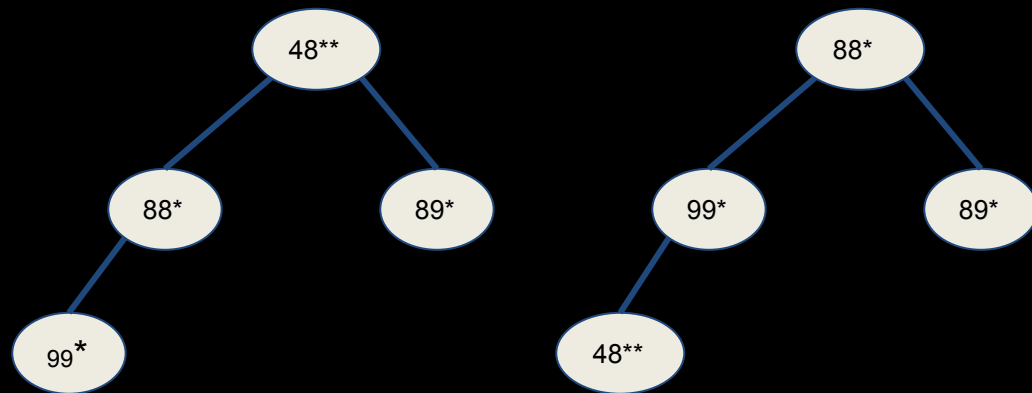


Arq 1: 6 3 7 11 12 15 29 35 42 45 55 65 76 89

Arq 2: 4* 8* 22* 23* 24* 45* 76*

Intercalação com seleção por substituição

6	12	15	11	3	7	29	35	42	55	45	65	76	89	8	4	22	24	23	45	89	99	88	76	48	78	32	1	2	20	30	16
---	----	----	----	---	---	----	----	----	----	----	----	----	----	---	---	----	----	----	----	----	----	----	----	----	----	----	---	---	----	----	----



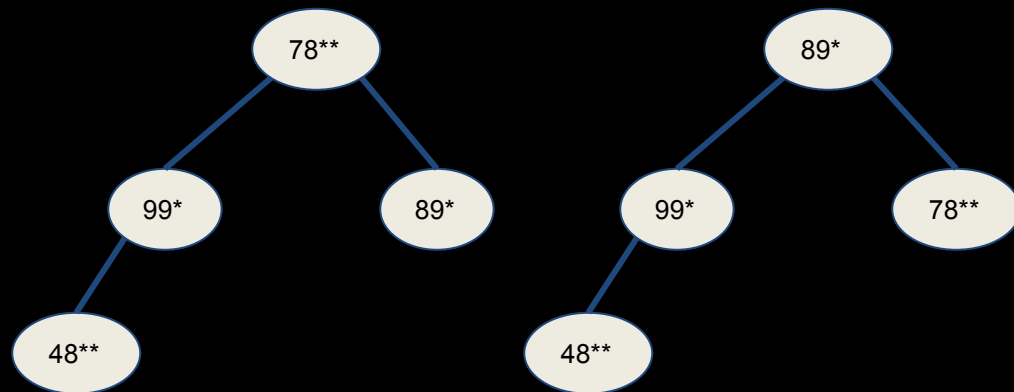
Arq 1: 6 3 7 11 12 15 29 35 42 45 55 65 76 89

Arq 2: 4* 8* 22* 23* 24* 45* 76* 88*

48 < 76, então vamos colocar uma marcação de novo bloco para 48, será 48**

Intercalação com seleção por substituição

6	12	15	11	3	7	29	35	42	55	45	65	76	89	8	4	22	24	23	45	89	99	88	76	48	78	32	1	2	20	30	16
---	----	----	----	---	---	----	----	----	----	----	----	----	----	---	---	----	----	----	----	----	----	----	----	----	----	----	---	---	----	----	----



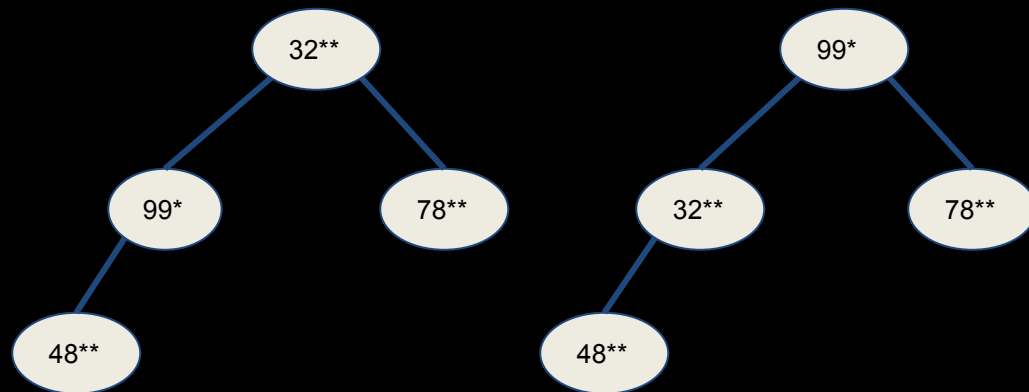
Arq 1: 6 3 7 11 12 15 29 35 42 45 55 65 76 89

Arq 2: 4* 8* 22* 23* 24* 45* 76* 88* 89*

78<88, então vamos colocar uma marcação de novo bloco para 78, será 78**

Intercalação com seleção por substituição

6	12	15	11	3	7	29	35	42	55	45	65	76	89	8	4	22	24	23	45	89	99	88	76	48	78	32	1	2	20	30	16
---	----	----	----	---	---	----	----	----	----	----	----	----	----	---	---	----	----	----	----	----	----	----	----	----	----	----	---	---	----	----	----



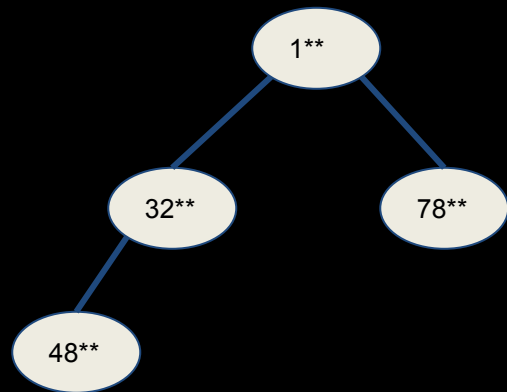
Arq 1: 6 3 7 11 12 15 29 35 42 45 55 65 76 89

Arq 2: 4* 8* 22* 23* 24* 45* 76* 88* 89* 99*

32<89, então vamos colocar uma marcação de novo bloco para 32, será 32**

Intercalação com seleção por substituição

6	12	15	11	3	7	29	35	42	55	45	65	76	89	8	4	22	24	23	45	89	99	88	76	48	78	32	1	2	20	30	16
---	----	----	----	---	---	----	----	----	----	----	----	----	----	---	---	----	----	----	----	----	----	----	----	----	----	----	---	---	----	----	----



Arq 1: 6 3 7 11 12 15 29 35 42 45 55 65 76 89 1**

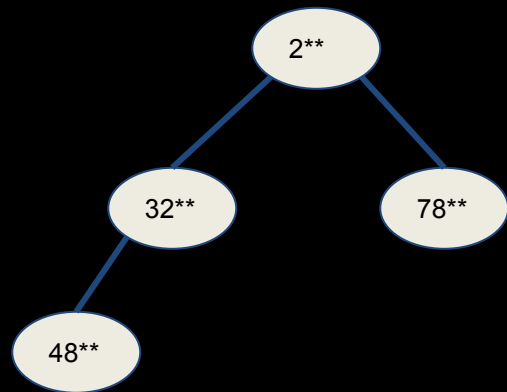
Arq 2: 4* 8* 22* 23* 24* 45* 76* 88* 89* 99*

1<99, então vamos colocar uma marcação de novo bloco para 1, será 1**

Note que todas as chaves no heap são do bloco
=> ao retirar 1 ela deverá ir para o arquivo 1

Intercalação com seleção por substituição

6	12	15	11	3	7	29	35	42	55	45	65	76	89	8	4	22	24	23	45	89	99	88	76	48	78	32	1	2	20	30	16
---	----	----	----	---	---	----	----	----	----	----	----	----	----	---	---	----	----	----	----	----	----	----	----	----	----	----	---	---	----	----	----

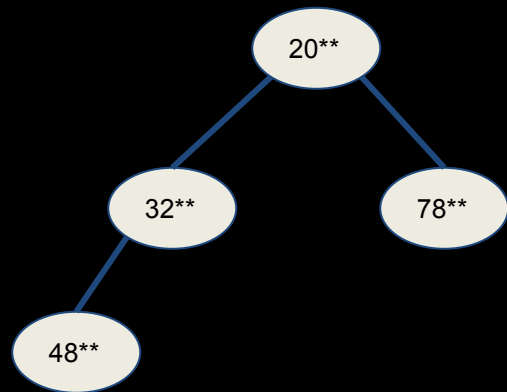


Arq 1: 6 3 7 11 12 15 29 35 42 45 55 65 76 89 1** 2***

Arq 2: 4* 8* 22* 23* 24* 45* 76* 88* 89* 99*

Intercalação com seleção por substituição

6	12	15	11	3	7	29	35	42	55	45	65	76	89	8	4	22	24	23	45	89	99	88	76	48	78	32	1	2	20	30	16
---	----	----	----	---	---	----	----	----	----	----	----	----	----	---	---	----	----	----	----	----	----	----	----	----	----	----	---	---	----	----	----

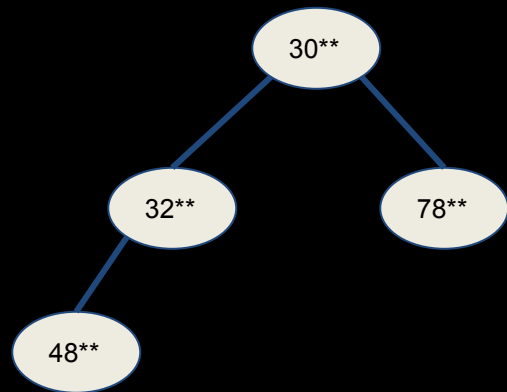


Arq 1: 6 3 7 11 12 15 29 35 42 45 55 65 76 89 1** 2**20**

Arq 2: 4* 8* 22* 23* 24* 45* 76* 88* 89* 99*

Intercalação com seleção por substituição

6	12	15	11	3	7	29	35	42	55	45	65	76	89	8	4	22	24	23	45	89	99	88	76	48	78	32	1	2	20	30	16
---	----	----	----	---	---	----	----	----	----	----	----	----	----	---	---	----	----	----	----	----	----	----	----	----	----	----	---	---	----	----	----

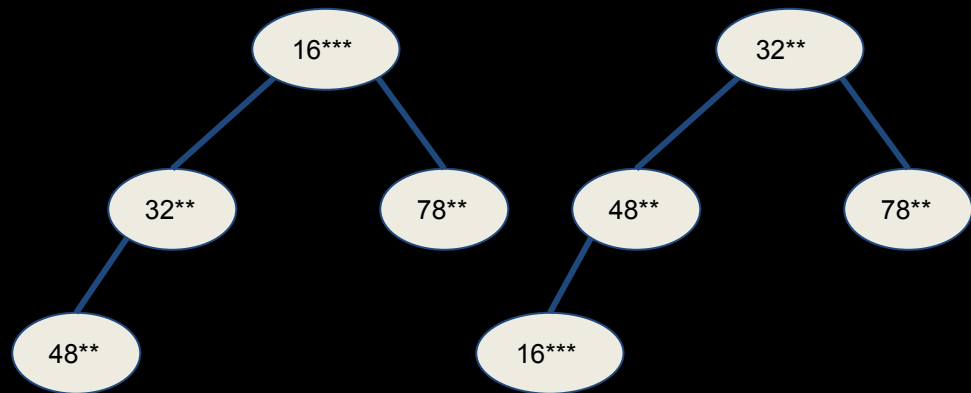


Arq 1: 6 3 7 11 12 15 29 35 42 45 55 65 76 89 1** 2**20**30**

Arq 2: 4* 8* 22* 23* 24* 45* 76* 88* 89* 99*

Intercalação com seleção por substituição

6	12	15	11	3	7	29	35	42	55	45	65	76	89	8	4	22	24	23	45	89	99	88	76	48	78	32	1	2	20	30	16
---	----	----	----	---	---	----	----	----	----	----	----	----	----	---	---	----	----	----	----	----	----	----	----	----	----	----	---	---	----	----	----



Arq 1: 6 3 7 11 12 15 29 35 42 45 55 65 76 89 1**2**20**30**32**

Arq 2: 4* 8* 22* 23* 24* 45* 76* 88* 89* 99*

16<30, então vamos colocar uma marcação de novo bloco para 16, será 16***

Acabaram as chaves do arquivo original => esvaziar o heap

Intercalação com seleção por substituição

6	12	15	11	3	7	29	35	42	55	45	65	76	89	8	4	22	24	23	45	89	99	88	76	48	78	32	1	2	20	30	16
---	----	----	----	---	---	----	----	----	----	----	----	----	----	---	---	----	----	----	----	----	----	----	----	----	----	----	---	---	----	----	----

Arq 1: 6 3 7 11 12 15 29 35 42 45 55 65 76 89 1**2**20**30**32**48**78**

Arq 2: 4* 8* 22* 23* 24* 45* 76* 88* 89* 99* 16***

No lugar de gerar 8 blocos ordenados usando a abordagem original, foram gerados apenas 4 blocos ordenados.