CI1056: Algoritmos e Estruturas de Dados II

Prof. Dr. Marcos Castilho

Departamento de Informática/UFPR

4 de dezembro de 2020

Resumo

Construção de um *heap*

Objetivos da aula

- Apresentar o algoritmo para construção de um *heap* de máximo

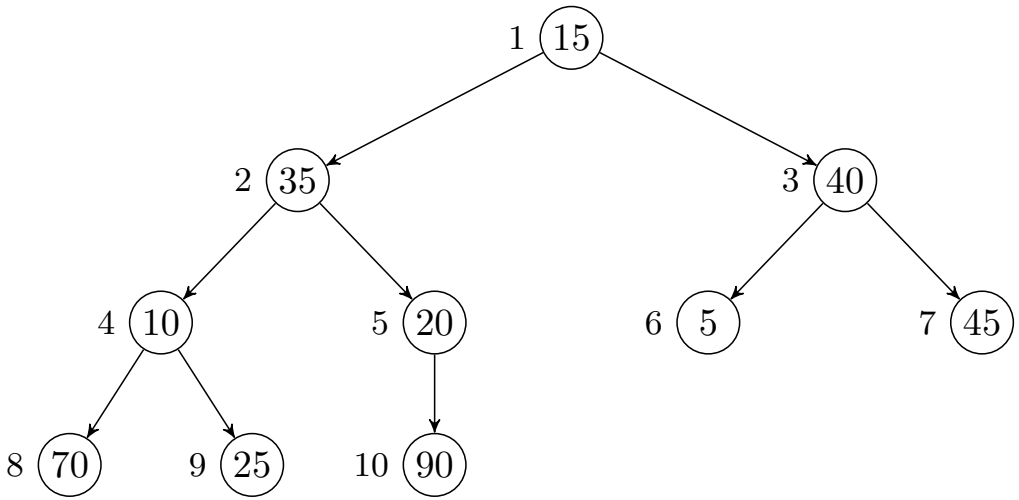
Motivação

- Como sabemos, um *heap* pode ser representado por um vetor
- Também veremos como um algoritmo de ordenação pode ser construído com base em um *heap* de máximo
- Logo, precisamos saber como transformar um vetor qualquer em um *heap* de máximo

Construção de um *heap* de máximo

- Dado um vetor qualquer, como ele pode ser visto na forma de uma árvore binária quase completa?

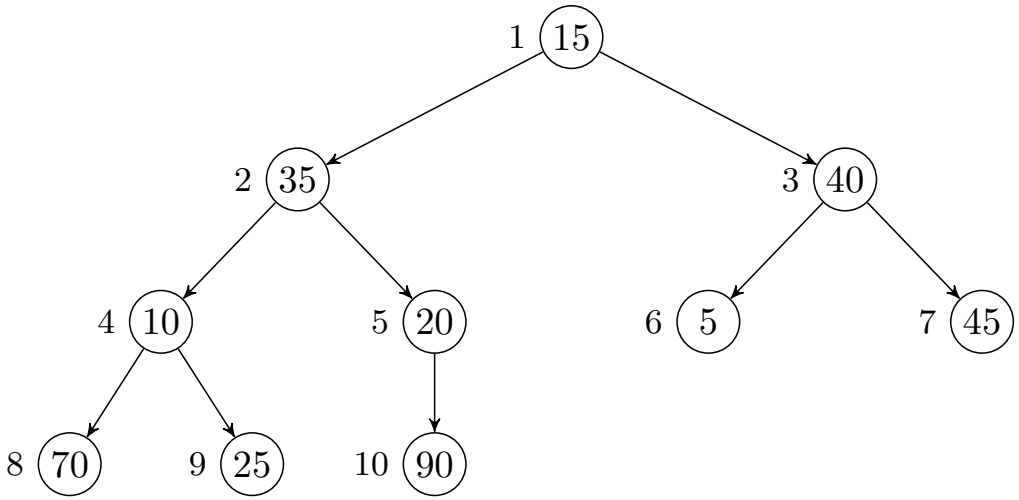
	1	2	3	4	5	6	7	8	9	10
v	15	35	40	10	20	5	45	70	25	90



Construção de um *heap* de máximo

- Isto não é um *heap* de máximo, é um vetor qualquer

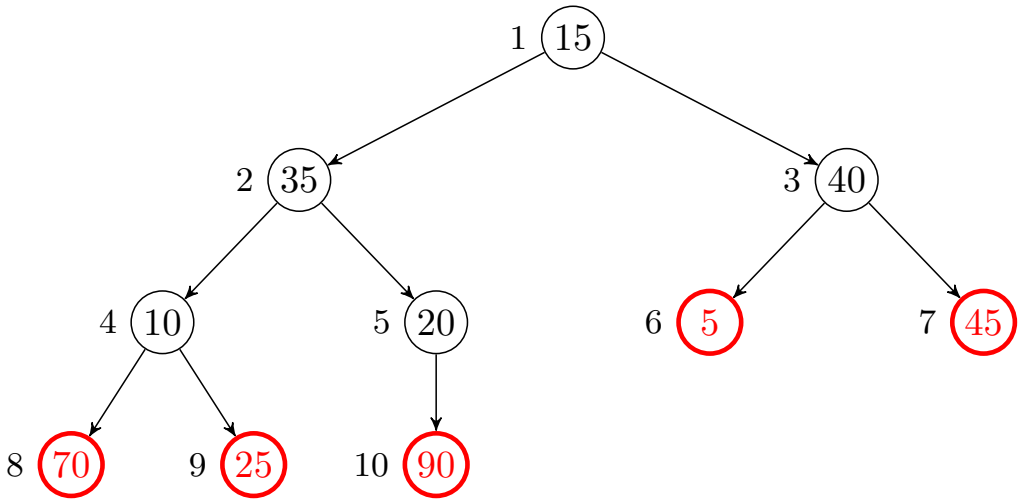
	1	2	3	4	5	6	7	8	9	10
v	15	35	40	10	20	5	45	70	25	90



Construção de um *heap* de máximo

- Observe cada nodo das folhas individualmente
- Cada folha é raiz de uma subárvore sem descendentes
- Logo, cada uma delas é um *heap* de máximo!

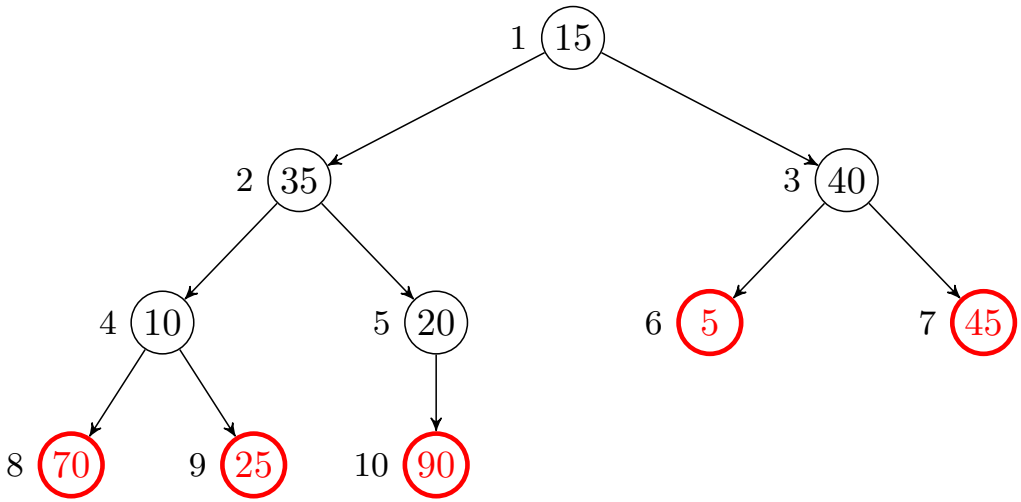
	1	2	3	4	5	6	7	8	9	10
v	15	35	40	10	20	5	45	70	25	90



Construção de um *heap* de máximo

- No vetor, isso corresponde aos índices de 6 em diante
- Isto é, a partir da metade + 1 do vetor

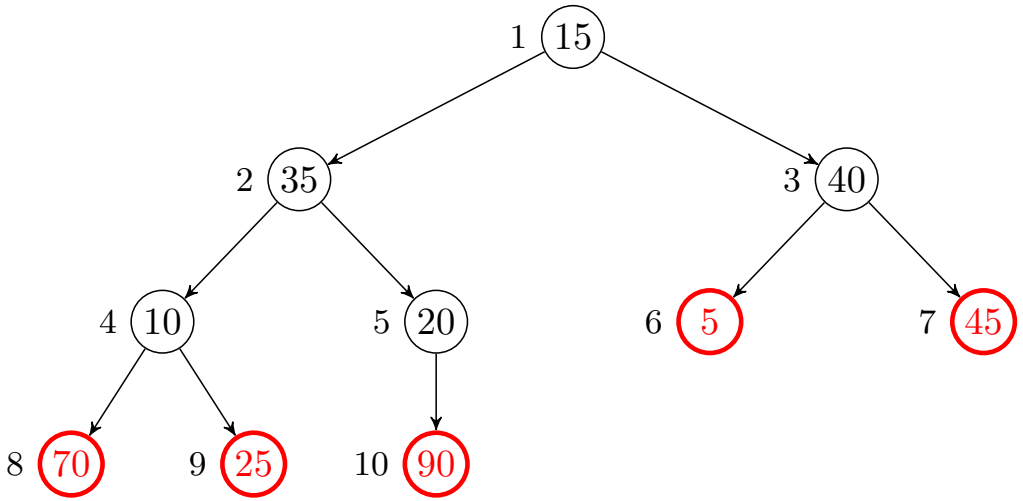
	1	2	3	4	5	6	7	8	9	10
v	15	35	40	10	20	5	45	70	25	90



Construção de um *heap* de máximo

- Observe atentamente os nodos rotulados 3, 4 e 5
- São nodos que tem como filhos raízes de *heaps* de máximo
- Eles satisfazem a condição de entrada do algoritmo `max_heapify`

	1	2	3	4	5	6	7	8	9	10
v	15	35	40	10	20	5	45	70	25	90



O algoritmo *buildheap*

- O algoritmo *build_max_heap* explora este fato
- De trás para frente, a partir da metade do vetor, aplica `max_heapify`
- Ao final, teremos um *heap* de máximo

Make_max_heap: construção de um *heap* de máximo

- Instância: (v, n), onde v é um vetor qualquer e n é o tamanho do vetor $v[1..n]$
- Resposta: retorna um (v, n) de forma que v é um *heap* de máximo

O algoritmo *build_max_heap*

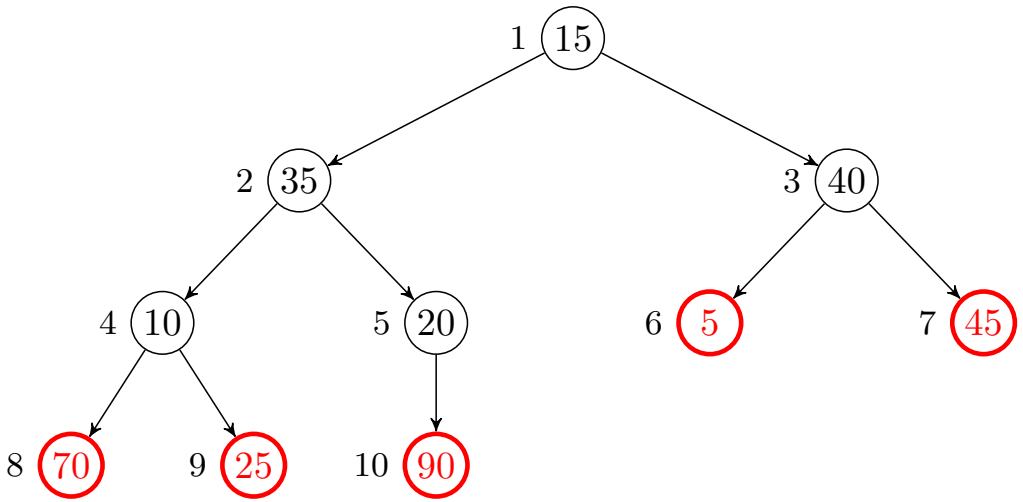
```
build_max_heap (v, n)
    para i de n/2 ate 1, regressivamente faca
        max_heapify (v, i)
```

- A ideia é muito simples e o algoritmo é iterativo
- Ele opera na primeira metade do vetor que tem $\lfloor n/2 \rfloor$ elementos
- Inicia pelo índice $\lfloor n/2 \rfloor$ e itera regressivamente até o nodo 1, isto é, até a raiz
- A cada itereção, a propriedade de *heap* de máximo é garantida, pois as duas subárvores de cada nodo são garantidamente *heaps* de máximo, justamente pois as folhas são trivialmente *heaps* de máximo
- O algoritmo `max_heapify` garante que a propriedade é estendida para o pai a cada iteração

Exemplo

- Na primeira iteração, o algoritmo `max_heapify` é aplicado no nodo 5

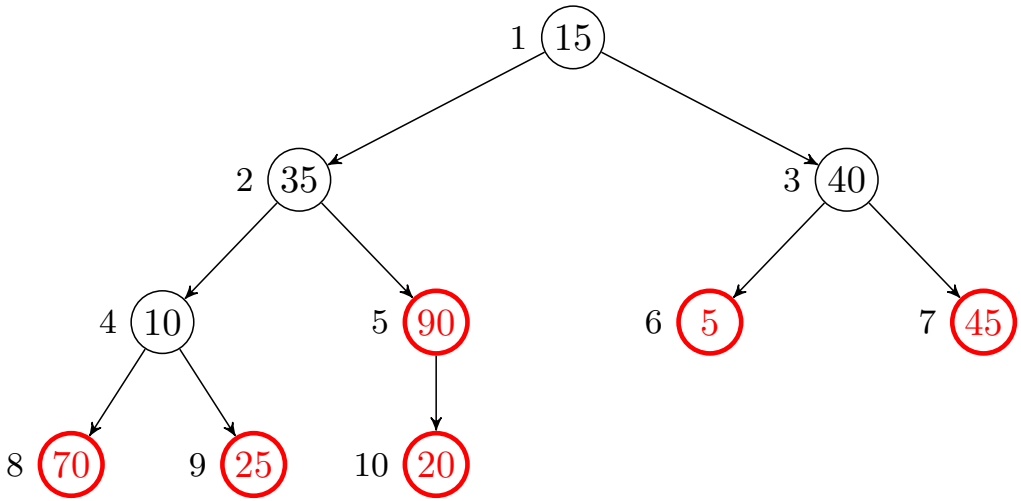
	1	2	3	4	5	6	7	8	9	10
v	15	35	40	10	20	5	45	70	25	90



Exemplo

- O resultado é este:

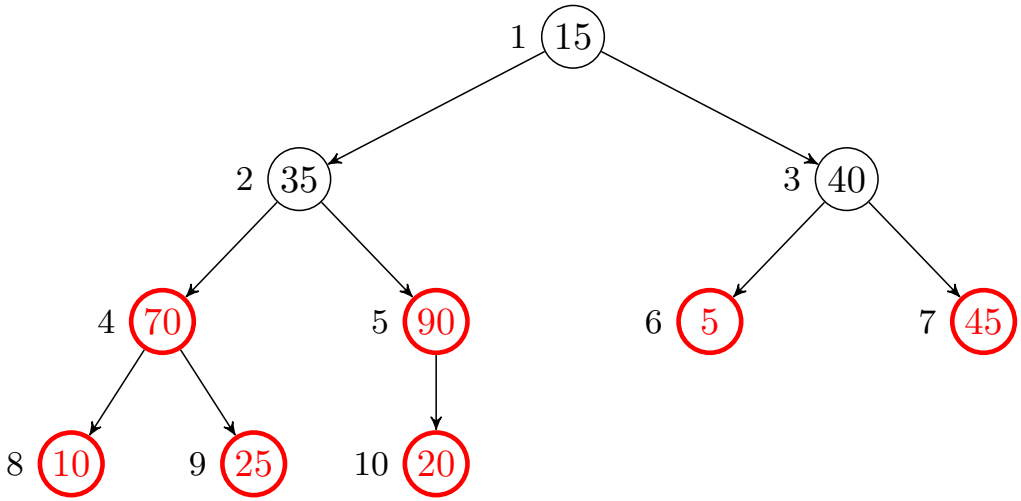
	1	2	3	4	5	6	7	8	9	10
v	15	35	40	10	90	5	45	70	25	20



Exemplo

- Em seguida, aplica `max_heapify` no nodo 4, obtendo:

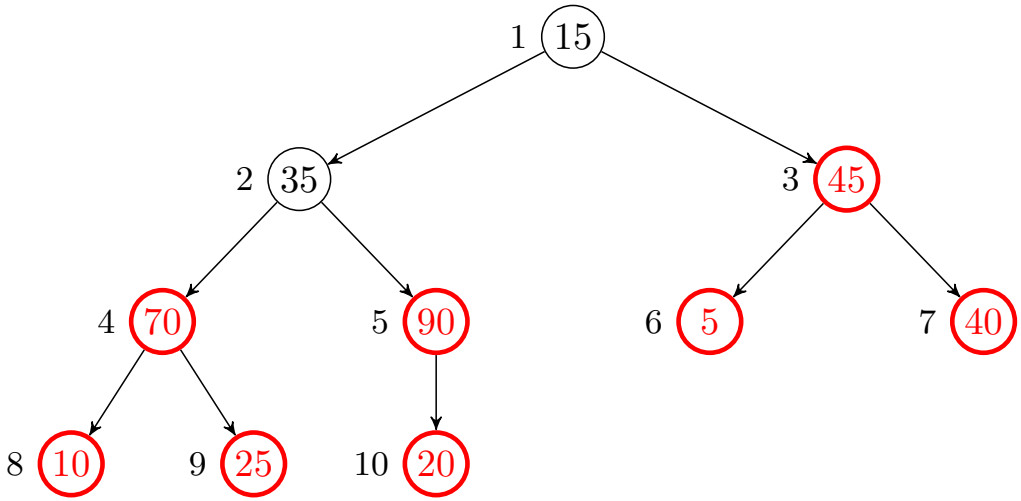
	1	2	3	4	5	6	7	8	9	10
v	15	35	40	70	90	5	45	10	25	20



Exemplo

- Em seguida, aplica `max_heapify` no nodo 3, obtendo:

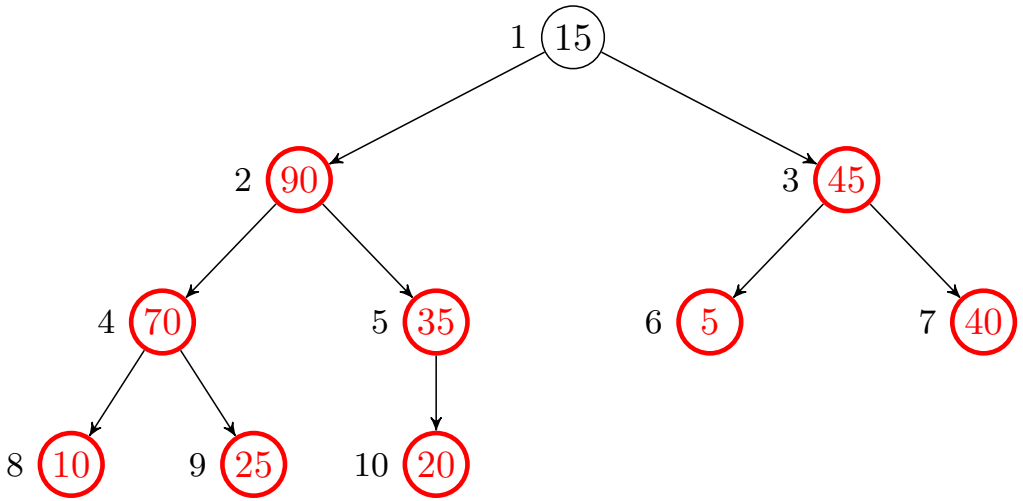
	1	2	3	4	5	6	7	8	9	10
v	15	35	45	70	90	5	40	10	25	20



Exemplo

- Em seguida, aplica `max_heapify` no nodo 2, obtendo:
- Observe que o 90 foi trocado com o 35 e o algoritmo parou na recursão seguinte, não tendo ido até a folha, pois o 35 já ficou no lugar dele

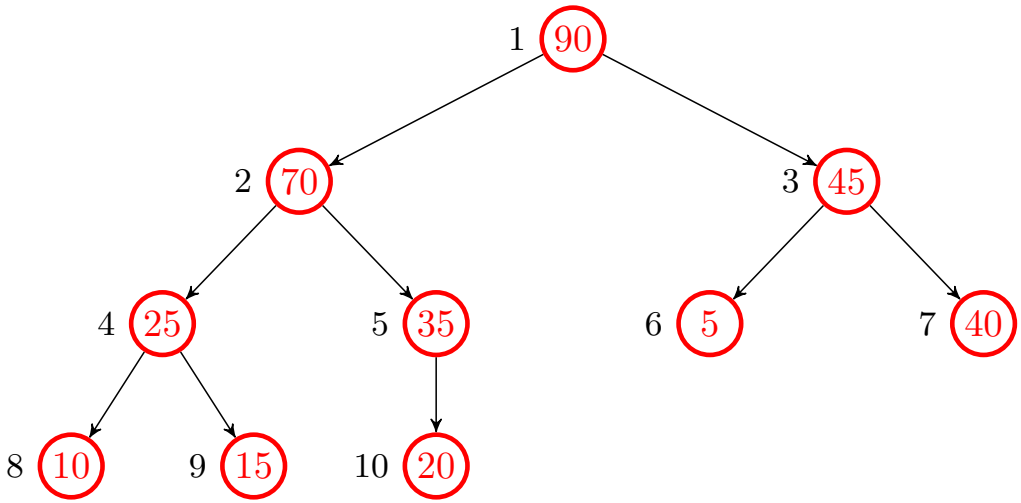
	1	2	3	4	5	6	7	8	9	10
v	15	90	45	70	35	5	40	10	25	20



Exemplo

- Finalmente, aplica `max_heapify` no nodo 1, obtendo um vetor que é um *heap* de máximo:
- Este foi um caso bem ruim, o 15 que estava na raiz foi parar na folha de índice 9, mas notem que custou 4 recursões, $(\lfloor log_2(10) \rfloor + 1)$

	1	2	3	4	5	6	7	8	9	10
v	90	70	45	25	35	5	40	10	15	20



Fim do tópico

- O conteúdo desta aula está no livro Cormen, Leiserson, Rivest e Stein, no capítulo 6, seção 6.3.

Licença

- Slides feitos em LaTeX usando beamer
- Licença

  Creative Commons Atribuição-Uso Não-Comercial-Vedada a Criação de Obras Derivadas 2.5 Brasil License.`http://creativecommons.org/licenses/by-nc-nd/2.5/br/`

  Creative Commons Atribuição-Uso Não-Comercial-Vedada a Criação de Obras Derivadas 2.5 Brasil License.`http://creativecommons.org/licenses/by-nc-nd/2.5/br/`