Heap Máximo - Perguntas e Respostas

---

1. Quais são os números mínimo e máximo de elementos em um heap de altura h?

- O número mínimo de elementos é `2^h`, que ocorre quando o heap está o mais desequilibrado possível.
- O número máximo de elementos é `2^(h+1) - 1`, que ocorre quando o heap é uma árvore binária completa.

---

2. Mostre que um heap de n elementos tem altura lg(n).

- Como um heap é uma árvore binária completa, seu número de nós cresce exponencialmente em função da altura.
- A relação entre o número de nós e a altura h é `n \approx 2^{h+1} - 1`.
- Resolvendo `h = log(n)`, obtemos `h = O(log n)`.

---

3. Mostre que, em qualquer subárvore de um heap de máximo, a raiz da subárvore contém o maior valor que ocorre em qualquer lugar nessa subárvore.

- Por definição, um heap de máximo exige que o pai seja maior ou igual a seus filhos.
- Logo, qualquer raiz de uma subárvore é maior ou igual a todos os valores abaixo dela.

---

4. Em que lugar do heap de máximo o menor elemento poderia residir, considerando que todos os elementos sejam distintos?

- O menor elemento estará sempre em uma das folhas, pois os elementos maiores estão concentrados no topo.

---

5. Um arranjo que está em sequência ordenada é um heap de mínimo?

- Se o arranjo estiver ordenado de forma crescente, ele satisfaz a propriedade de um **heap de mínimo**.
- No entanto, um arranjo ordenado não segue necessariamente a estrutura de uma árvore binária completa, logo não pode ser diretamente um heap.

---

6. A sequência (23, 17, 14, 6, 13, 10, 1, 5, 7, 12) é um heap de máximo?

- Não, pois em um heap de máximo, cada pai deve ser maior que seus filhos.
- Neste caso, `6` é pai de `13`, que é maior que ele, violando a propriedade de heap de máximo.

---

7. Mostre que, com a representação de arranjo para ordenar um heap de n elementos, as folhas são os nós indexados por n/2 + 1, n/2 + 2, ..., n.

- Em um heap armazenado em um array, os filhos de um nó `i` estão nos índices `2i` e `2i+1`.
- Todos os nós a partir de `n/2 + 1` não têm filhos, portanto, são folhas.

---

8. Por que queremos que o índice de laço i na linha 2 de BUILD-MAX-HEAP diminua de A*comprimento/2 até 1, em vez de aumentar de 1 até A*comprimento/2?

- Para garantir que os elementos de maior profundidade sejam corrigidos antes dos elementos superiores.
- Isso evita a necessidade de reajustar elementos múltiplas vezes.

---

9. Mostre que existem, no máximo, n/2^h + 1 nós de altura h em qualquer heap de n elementos.

- A árvore binária completa segue a distribuição exponencial dos nós por nível.
- O número de nós no nível h é no máximo `n/2^h + 1`, pois os nós diminuem conforme a profundidade aumenta.

---

10. Qual é o efeito de chamar MAX-HEAPIFY(A, i) quando o elemento A[i] é maior que seus filhos?

- Nenhuma modificação ocorre, pois a propriedade de heap já está satisfeita.

---

11. Qual é o efeito de chamar MAX-HEAPIFY(A, i) para i > A · tamanho-do-heap/2?

- Nenhum efeito ocorre, pois `i` é um índice de uma folha, e folhas já satisfazem a propriedade de heap.

---

12. Escreva um MAX-HEAPIFY eficiente que use um laço em vez de recursão.

- O algoritmo pode ser implementado usando um laço `while`, garantindo a reorganização sem chamadas recursivas.

---

13. Mostre que o tempo de execução do pior caso de MAX-HEAPIFY para um heap de tamanho n é W(lg n).

- No pior caso, o algoritmo desce até a última camada da árvore, percorrendo `O(log n)` níveis.

---

14. Ilustre a operação de MAX-HEAPIFY(A, 3) sobre o arranjo A = ⟨27, 17, 3, 16, 13, 10, 1, 5, 7, 12, 4, 8, 9, 0⟩.

- A transformação ocorre deslocando elementos para manter a propriedade de heap, conforme ilustrado na figura do livro.

---

15. Escreva o pseudocódigo para MIN-HEAPIFY(A, i) e compare seu tempo de execução com MAX-HEAPIFY.

- `MIN-HEAPIFY` é similar a `MAX-HEAPIFY`, mas ajusta para manter um heap de mínimo.
- Seu tempo de execução é também `O(log n)`, pois a operação percorre no máximo a altura do heap.

---

16. Ilustre a operação de BUILD-MAX-HEAP no arranjo A = ⟨5, 3, 17, 10, 84, 19, 6, 22, 9⟩.

- O processo ajusta os nós internos para transformar o arranjo em um heap de máximo.