TP 06

Ana Neri

17 de Abril 2023

Exercícios práticos sobre estruturas de dados.

Exercício 1 Em Python existe um tipo de dados chamado `Set`.

O conjunto (`Set`) é coleção não ordenada, imutável (é possível remover e adicionar items) e não indexada.

1. A duplicações não são permitidas. Por exemplo, pode ver como funciona executando o seguinte conjunto:

   ```
   thisset = {"ola", "amendoas", "chocolates", "ola"}
   ```

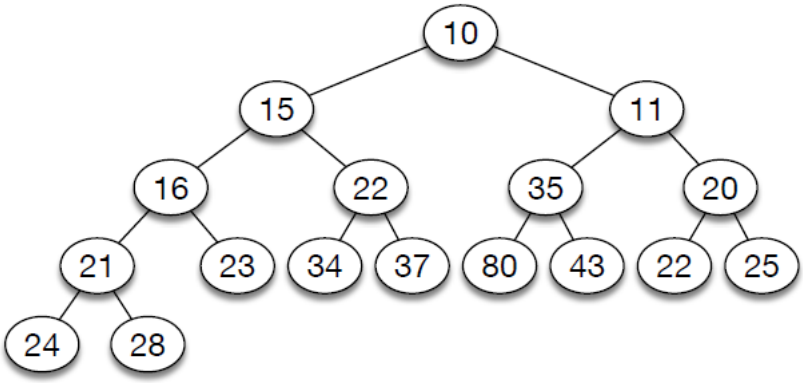
   Explique o que acontece quando tenta implementar um conjunto com

   ```
   thatset = {"ola", 2, True, 1, "chocolate"}
   ```

2. Faça a interseção de dois sets.

Exercício 2 Implemente com arrays estáticos, dinâmicos e listas ligadas a stack e a queue.

Exercício 3 Uma min-heap é uma árvore binária em que cada nodo é menor ou igual a todos os seus sucessores. Por outro lado, uma árvore diz-se semi-completa se todos os níveis da árvore estão completos, com a possível exceção do último, que pode estar parcialmente preenchido (da esquerda para a direita). As árvores semi-completas têm uma representação "económica" em array: os nodos são armazenados por nível, sempre da esquerda para a direita. Por exemplo, a árvore (que é uma min-heap)



Que pode ser guardado no array:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
10	15	11	16	22	35	20	21	23	34	37	80	43	22	25	24	28

1. Analisando o exemplo acima, determine expressões gerais que permitam calcular:
   (a) O índice onde se encontra a sub-árvore esquerda do nodo da posição i.
   (b) O índice onde se encontra a sub-árvore direita do nodo da posição i.
   (c) O índice onde se encontra o pai do nodo da posição i.
   (d) O índice onde se encontra a primeira folha, i.e., o primeiro nodo que não tem sucessores.

2. Defina a função `bubbleUp (i, h)` que (por sucessivas trocas com o pai) puxa o elemento que está na posição `i` da min-heap `h` até que satisfaçaa a propriedade das min-heaps.

   Identifique o pior caso desta função e calcule o número de comparações/trocas efetuadas nesse caso.

3. Defina a função `bubbleDown ( i, h, N)` que (por sucessivas trocas com um dos filhos) empurra o elemento que está na posição `i` da min-heap `h` até que satisfaça a propriedade das min-heaps.

   Identifique o pior caso desta função e calcule o número de comparações/trocas efetuadas nesse caso.

4. Defina uma função `ordenaHeap (h, N)` que, usando a função `bubbleDown` definida acima, transforma a min-heap `h`, num array ordenado por ordem decrescente.