

Exercícios

Visite o repositório a seguir para obter uma cópia da solução em que você vai basear as soluções dos exercícios propostos.

https://github.com/professorbossini/20251_maua_cic401

1 Os algoritmos a seguir envolvem o uso de uma estrutura de dados Heap para fazer a implementação de uma fila de prioridades, tal como descrito no livro do Cormen. Faça a sua implementação em Java.

HEAP-MAXIMUM(A)

1 **return** $A[1]$

HEAP-EXTRACT-MAX(A)

```

1  if  $A.heap-size < 1$ 
2      error “heap underflow”
3   $max = A[1]$ 
4   $A[1] = A[A.heap-size]$ 
5   $A.heap-size = A.heap-size - 1$ 
6  MAX-HEAPIFY( $A, 1$ )
7  return  $max$ 
```

HEAP-INCREASE-KEY(A, i, key)

```

1  if  $key < A[i]$ 
2      error “new key is smaller than current key”
3   $A[i] = key$ 
4  while  $i > 1$  and  $A[PARENT(i)] < A[i]$ 
5      exchange  $A[i]$  with  $A[PARENT(i)]$ 
6       $i = PARENT(i)$ 
```

MAX-HEAP-INSERT(A, key)

1 $A.heap-size = A.heap-size + 1$

2 $A[A.heap-size] = -\infty$

3 HEAP-INCREASE-KEY($A, A.heap-size, key$)

Referências

BONDY, J. A.; MURTY, U. S. R. **Graph theory**. New York: Springer, 2008. (Graduate Texts in Mathematics, v. 244).

CORMEN, Thomas H. et al. **Introduction to Algorithms**. 3. ed. Cambridge: MIT Press, 2009.

FEOFILOFF, Paulo. **Análise de Algoritmos**. Disponível em: https://www.ime.usp.br/~pf/analise_de_algoritmos/. Acesso em: março de 2025.

KLEINBERG, Jon; TARDOS, Éva. **Algorithm Design**. Boston: Pearson, 2006.