**CI056: ALGORITMOS E ESTRUTURAS DE DADOS II**

# Heap, Heap, UPA!

**Trabalho Prático 2 (2022)**

## Introdução

"Heap, heap: UPA!" tem sido muito importante na comunidade no que diz respeito ao atendimento e resposta a emergências médicas. É o sistema que permite enfileirar e retirar os pacientes quando são levados ao Pronto Socorro. O sistema enfileira os pacientes com prioridade de "atenção urgente necessária" para "atenção necessária". Toda a documentação que abrange todo o projeto, desde sua conceituação até sua finalização, pode ser encontrada no site https://www.inf.ufpr.br/kkq20/t2/trabalho.html .

## Objetivos

- O sistema deve permitir que Madame Estraela cadastre os pacientes que chegam à UPA, que chame o próximo que vai ser atendido.
- As operações que devem ser disponibilizadas incluem: InicHeap, InsereHeap, RemoveHeap, Heapfy, ChecaHeap, ImprimeHep e HeapSort.
- O sistema deve imprimir todos na sala de espera, que ordena todos os elementos de acordo com sua prioridade.
- Atualizar a prioridade de algum paciente que piorou ou melhorou.
- A qualquer momento deve ser possível imprimir o Heap.

## Cronologia do Projeto

Este projeto teve início em 19/08/2022 e foi finalizado em 01/09/2022. A cronologia precisa dos eventos que levaram a este projeto a se concretizar também pode ser encontrada em https://www.inf.ufpr.br/kkq20/t2/Relatorio/WorkLog.txt .

## Implementação do Projeto

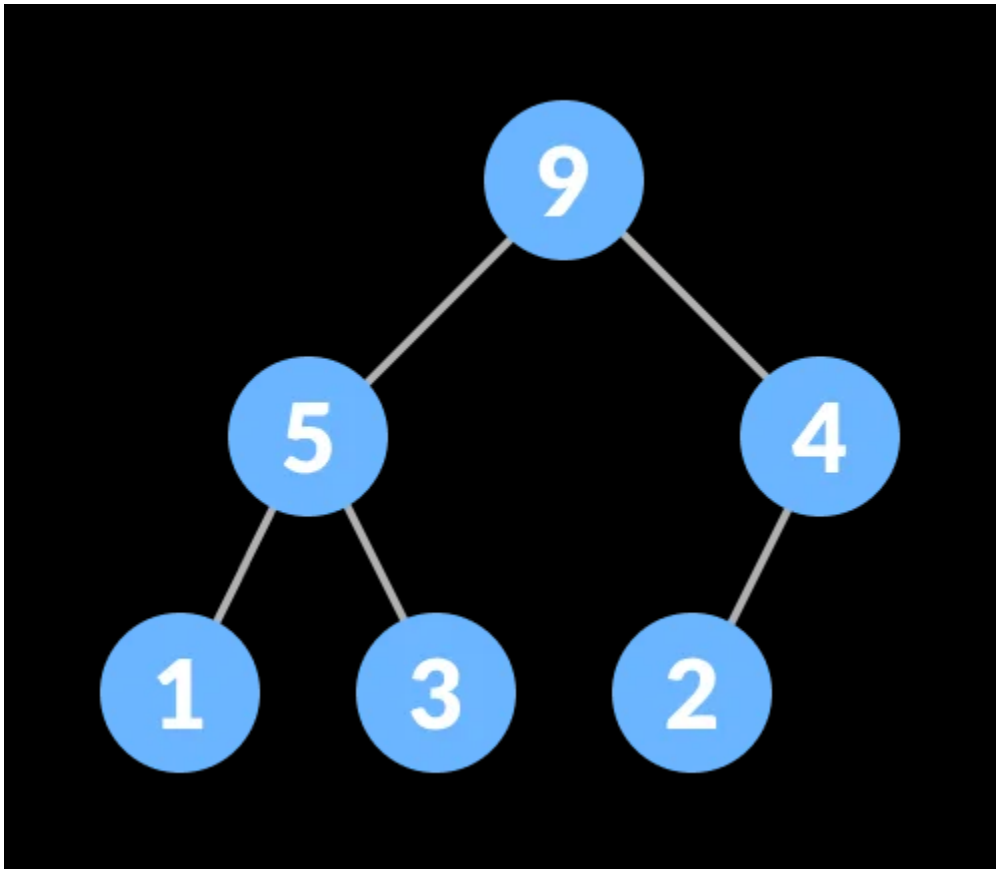
Para implementar o sistema, primeiro criamos uma estrutura contendo o nome e a prioridade de cada paciente.

Em seguida, criamos um heap para conter a estrutura.

Criamos então várias funções para permitir que enfermeiros e profissionais médicos manipulem as pilhas e as prioridades dos pacientes. Entre essas funções estão:

- **InicHeap** : Esta função inicia a construção de um heap.
- **FreeHeap** : Esta função remove todos os elementos, neste caso pacientes, de um heap.

- **InsereHeap** : Esta função permite a inserção de um paciente no heap sem prejudicar a prioridade dos pacientes já enfileirados. Ele faz isso primeiro verificando se o heap está cheio ou não. Se o heap estiver cheio, ele informa ao usuário. Se não for, inerte o paciente em uma posição adequada em relação à sua prioridade.
- **RemoveHeap** : Esta função permite a remoção de um paciente da pilha sem prejudicar a prioridade dos pacientes já enfileirados. Ele faz isso verificando primeiro se o heap está vazio ou não. Se o heap estiver vazio, ele informa ao usuário. Se não for, ele remove o paciente desejado e reorganiza o heap em ordem de prioridade.
- **Heapfy** : Esta é a função chave na organização dos pacientes de acordo com a prioridade. Os pacientes são organizados de tal forma que os casos mais urgentes são colocados no topo da pilha. É o processo de criar uma estrutura de dados heap a partir de uma árvore binária. Conforme exemplificado na Figura 1.0




- **BuildHeap** : Esta função usa o Heapfy para organizar todos os pacientes no conjunto/vetor em um heap.

- **ChecaHeap** : Esta função verifica o heap, se está vazio ou não, e mostra o próximo paciente em ordem de prioridade
- **ImprimeHeap** : Esta função mostra todos os pacientes na árvore binária completa do heap criado.
- **HeapSort** : Esta função ordena os pacientes no heap com respeito colocando os pacientes com maior prioridade no topo, que no nosso caso é o Max-heap.
- **AlteraHeap** : Esta função altera a prioridade de um paciente na pilha. A função busca no vetor de pacientes de forma linear para encontrar o paciente cuja prioridade deve ser alterada. Se encontrado, a prioridade desse elemento é alterada e o restante do heap é reorganizado com o Heapfy.
- **EncheHeap** : Esta função cria um número aleatório de pacientes solicitados. Observe que ele os coloca em um heap, mas não de maneira ordenada.

## Conclusão

Atender os pacientes levando em consideração sua prioridade de necessidade é uma maneira muito eficiente de administrar um equipamento de saúde e esse sistema ajudará bastante a atingir esse nível de eficiência. Os pacientes que precisam de atenção urgente sempre receberão a atenção de que precisam.