

AULA 11 - FILAS COM PRIORIDADE

Comece por ler o Capítulo 9 – Filas com prioridade, mais concretamente o item 9.4 – Implementação com amontoado (páginas 439-443), para se familiarizar com a implementação de uma fila com prioridade baseada num amontoado binário (*binary heap*) e os respetivos algoritmos.

O tipo de dados abstrato `PQUEUE_HEAP` é constituído pelo ficheiro de interface `pqueue.h` e pelo ficheiro de implementação `pqueue.c` e implementa a manipulação de filas com prioridade baseadas em amontoados binários, orientadas aos máximos e que armazenam números inteiros, com capacidade de múltipla instanciação e cujas operações devolvem um código de erro relativo à sua execução.

- Comece por testar convenientemente toda funcionalidade do tipo de dados usando os programas `testpqueue.c`, `simpqueue.c` e a *makefile* `mkpqueue`.
- Implemente e teste uma versão do algoritmo **Heap-Sort** usando esta fila com prioridade.
- Dada uma sequência de **n** elementos inteiros (com **n** muito grande), pretende-se determinar as suas **k ocorrências** (com eventuais repetições) de **menor valor** (com $k \ll n$).
- Sem ordenar uma sequência dada, e usando esta fila com prioridade (mas **sem criar uma fila com prioridade de n elementos**), implemente uma função que lhe permita determinar as **k ocorrências de menor valor** de uma sequência dada.

TRABALHO OPCIONAL

Implemente o tipo de dados abstrato `MIN_PQ` para manipular filas com prioridade baseadas em amontoados binários, orientadas aos mínimos e que armazenam números inteiros, com capacidade de múltipla instanciação e cujas operações devolvem um código de erro relativo à sua execução.

- Comece por testar convenientemente toda funcionalidade do tipo de dados adaptando os programas `testpqueue.c` e `simpqueue.c`.
- Implemente e teste uma versão do algoritmo **Heap-Sort** usando esta fila com prioridade.
- Dada uma sequência de **n** elementos inteiros (com **n** muito grande), pretende-se agora determinar as suas **k ocorrências** (com eventuais repetições) de **maior valor** (com $k \ll n$).
- Sem ordenar uma sequência dada, e usando esta fila com prioridade (mas **sem criar uma fila com prioridade de n elementos**), implemente uma função que lhe permita determinar as **k ocorrências de maior valor** de uma sequência dada.

