

Trabalho 6 - A outra pilha

Joãozinho está planejando abrir um restaurante especializado em comida britânica e pediu sua ajuda para priorizar os pratos do cardápio. Mas você já implementou o Quicksort e refutou Joãozinho, o que fazer? Anda pensando muito na palavra “priorizar”, deve existir uma ligação entre “prioridade”, “filas” e “ordenação”. Fatalmente fez a pergunta ao ChatGPT, e ele lhe respondeu:

A relação entre filas, prioridade e ordenação se manifesta de forma clara em algoritmos como o **heapsort**, que utiliza uma estrutura de dados chamada **fila de prioridade** para organizar elementos com base em um critério definido, onde itens com maior prioridade são processados primeiro, garantindo uma ordenação eficiente dos dados.



Aproveitando que não terá aula na sexta — sua escola será local de votação — comparará esse a outro algoritmo, afinal, sua mente é binária.

Implementação

Implemente os métodos de ordenação Merge Sort e Heapsort, receberá $0 < k < 3 \cdot 10^5$ pratos.

Ordene em ordem crescente pelo inteiro que representa prioridade, se a prioridade for a mesma, o de maior tempo deverá ser o primeiro a ser impresso, ou seja, de forma decrescente. Nenhum prato tem a mesma prioridade e tempo de preparo.

A prioridade é um inteiro $0 < p < 3 \cdot 10^5$, o tempo de preparo (em minutos) é um inteiro $0 < t < 10^3$, o nome é uma string de no máximo 50 caracteres, sem espaço.

Exemplo

Entrada:

```
4
7 30 fish-and-chips
9 29 margarine-unie
9 60 steak-and-ale-pie
28 60 meat-heap
```

Saída:

```
fish-and-chips
steak-and-ale-pie
margarine-unie
meat-heap
```

Relatório

Compare os métodos de ordenação e faça uma breve explicação sobre a estrutura *heap* e a complexidade de espaço do Merge Sort.

Bônus de 1.0 no código:

Outra dúvida trespassa sua mente: qual a tradução de *heap*? Pilha?

Implemente o método Heapsort utilizando a *heap* no formato array e com ponteiros (como uma árvore binária). Escreva sobre a implementação no relatório e coloque *prints* do código.