

# Tutorial: Caching Offline

O problema consiste em simular um gerenciador de cache e determinar o número mínimo de falhas de página. Como a sequência completa de requisições é conhecida previamente, podemos utilizar uma estratégia gulosa para resolver o problema.

## Representação do estado

- A memória principal (cache) é representada por um conjunto (*hash set*), que armazena os identificadores das páginas atualmente carregadas, permitindo consultas em tempo constante.
- O histórico futuro de cada página é mapeado em vetores, registrando todos os índices (momentos no tempo) em que cada página será requisitada.
- Um vetor de ponteiros auxiliares acompanha qual a próxima ocorrência futura correspondente para cada página.

## Estratégia de solução usando Algoritmo Guloso

A estratégia ótima dita que, quando a cache estiver cheia e ocorrer uma falha, a página a ser substituída deve ser aquela que demorará mais tempo para ser referenciada novamente no futuro.

1. Leia a capacidade da cache ( $k$ ) e o tamanho da sequência de requisições ( $v$ ).
2. Faça um pré-processamento da sequência de páginas:
  - Armazene a sequência original em um vetor.
  - Para cada página lida, adicione o índice atual à sua respectiva lista de referências futuras.
3. Inicialize a cache vazia, o controle de progresso das referências futuras e o contador de falhas (*misses*).
4. Percorra a sequência de requisições passo a passo:
  - Se a página atual **não** estiver na cache, ocorreu uma falha de página. Incremente o contador de *misses*.
  - Se a cache já estiver cheia (tamanho igual a  $k$ ) no momento da falha, é necessário eleger uma página residente para remoção:
    - Inspeccione todas as páginas atualmente na cache.
    - Se alguma página não possuir mais requisições futuras, ela é a candidata ideal e deve ser escolhida para remoção imediata.
    - Caso contrário, compare a próxima requisição de todas e encontre aquela que vai demorar mais tempo para aparecer de novo (índice mais distante no futuro), removendo-a.
  - Insira a nova página na cache.
  - Ao final de cada iteração, independentemente de ter ocorrido acerto ou falha, avance o ponteiro de ocorrências da página que acabou de ser processada.
5. Ao término da simulação, imprima o total de falhas contabilizadas.