

Cap 3 - Questionário

Responda cada pergunta em 2-3 frases, utilizando as informações do capítulo do livro-texto.

martinson.freitas@gmail.com [Mudar de conta](#)



Rascunho salvo.

* Indica uma pergunta obrigatória

Enviar por e-mail *



Registrar **martinson.freitas@gmail.com** como o e-mail a ser incluído na minha resposta

1. Qual a principal desvantagem das listas encadeadas em comparação com implementações baseadas em arrays para as operações `get(i)` e `set(i,x)`? * 10 pontos

Listas encadeadas têm acesso ineficiente ($O(n)$) a posições arbitrárias (`get(i)/set(i,x)`), pois exigem percorrer os nós sequencialmente.

2. Qual a principal vantagem das listas encadeadas em comparação com implementações baseadas em arrays para as operações de adição ou remoção de um nó adjacente a um nó dado? * 10 pontos

Numa `SSList` as operações `add()` e `remove()` são executadas em um tempo $O(1)$ por operação. Em arrays essas operações exigem deslocar elementos com um custo de operação $O(n)$.



3. Como uma SLList implementa as operações de Stack (push() e pop()) * 10 pontos de forma eficiente?

As operações push() e pop() são implementadas de forma eficiente $O(1)$ usando o nó head; push(x) adiciona um nó no início, apontando-o para head; pop() remove o nó head e retorna seu valor, atualizando o head para o próximo nó.

4. Como uma SLList implementa as operações de Queue (add() e remove()) de forma eficiente? * 10 pontos

Uma SLList implementa as operações de Queue(add()/remove()) de forma eficiente $O(1)$ usando dois ponteiros: um para o início (head) e outro para o fim (tail) da lista. As remoções são feitas a partir do início e as adições são feitas no final da lista.

5. Por que é difícil remover o último nó (tail) de uma SLList eficientemente? * 10 pontos

Remover a cauda de uma SLList é difícil porque requer a atualização do valor de 'tail' para que ele aponte para o nó "w" que precede "tail", este nó "w" tal que $w.next = tail$. Essa operação tem o custo $O(n)$.

6. Qual o propósito do nó dummy em uma DLList? * 10 pontos

Este é um nó que não contém quaisquer dados, mas age como um espaço reservado para que não haja nós especiais; cada nó tem um next e um prev, com o dummy agindo como nó que sucede o último nó da lista e que precede o primeiro nó da lista, assim os nós da lista são ligados em um ciclo.



7. Qual o tempo de execução para as operações $\text{get}(i)$ e $\text{set}(i,x)$ em uma DLList, e o que domina esse tempo? * 10 pontos

O tempo de execução dessas operações é dominado pelo tempo que leva para encontrar o i -ésimo nó, $O(1 + \min\{i, n-1\})$.

8. Qual a desvantagem das listas encadeadas (SLList e DLList) em termos de uso de espaço? * 10 pontos

Listas encadeadas consomem mais memória que arrays devido ao overhead de ponteiros (1 ou 2 por nó). Cada nó exige armazenar valores + ponteiros, ocupando até 6x mais espaço que um array para a mesma quantidade de dados.

9. Como uma SEList aborda a desvantagem de espaço das listas encadeadas tradicionais? * 10 pontos

A SEList mitiga o problema de espaço das listas encadeadas tradicionais ao agrupar elementos em blocos (arrays), reduzindo o número de nós e ponteiros, mantendo operações eficientes ($O(\sqrt{n})$) com balanceamento entre blocos e otimizando o uso de memória sem sacrificar totalmente a flexibilidade de inserções/remoções.

10. Qual a principal propriedade de um BDeque que o torna útil como bloco em uma SEList? * 10 pontos

☒ Opção 1

Enviar

[Limpar formulário](#)

Nunca envie senhas pelo Formulários Google.

Este formulário foi criado em FEN UERJ.
Este formulário parece suspeito? [Relatório](#)



Google Formulários