

Exercícios de fixação - **Gerenciamento de memória**

Questão 1. Concorde ou discorde: “Usando uma lista de blocos livre, nunca ocorrerá fragmentação de memória se todos blocos nunca forem desalocados.” Justifique. Explique o que é fragmentação de memória e quando ela ocorre.

Questão 2. Uma função resolve um problema por retrocesso, de forma que a solução do problema é representada por um vetor de n posições. A primeira versão utilizava recursão explicitamente, mas só executava para instâncias pequenas. Para instâncias em que n era grande, ocorria *stack overflow*. Para resolver o problema, a função foi reimplementada usando uma pilha com lista encadeada e agora executa corretamente para instâncias de tamanho arbitrário. O curioso é que, para instâncias com n pequeno, a função anterior era muito mais rápida.

- (a) Disserte sobre o que é uma lista de blocos livres. Explique como ela pode afetar o tempo de execução de um programa.
- (b) Explique porque a primeira versão da função funcionava mais rapidamente para n pequeno.
- (c) Dê sugestões para corrigir esse problema.

Questão 3. Um programa tem como objetivo calcular e devolver o resultado de algumas simulações complexas que dependem dos parâmetros digitados; o programa é interativo e deve devolver a resposta tão logo os parâmetros forem digitados e a tecla ENTER for pressionada. Na maioria das vezes, o programa responde instantaneamente, mas algumas vezes, ele pode demorar alguns instantes (até mesmo segundos) antes de dar uma resposta.

- (a) Uma questão problemática da coleta de lixo com algoritmos baseados em marcação é que não há controle sobre quando ela é executada. Explique porque a seguinte situação hipotética acima poderia.
- (b) Sugira estratégias para amenizar o problema do programa acima. É possível garantir que isso nunca ocorrerá?

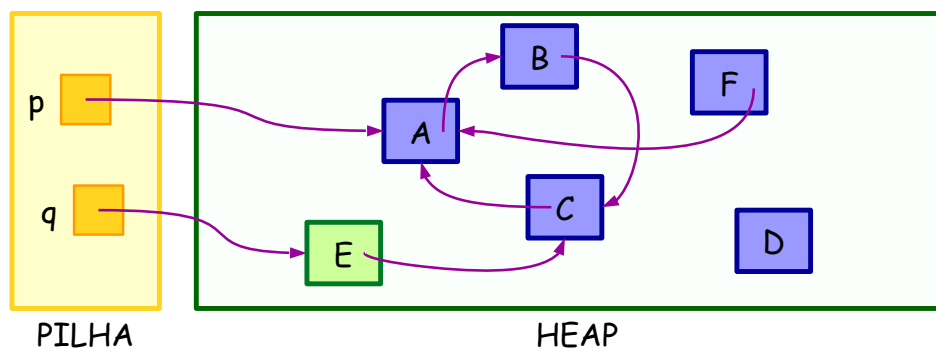
Questão 4. Liste o maior número de linguagens de programação que conseguir e, para cada uma, diga se a memória é gerenciada automaticamente e, quando for o caso, qual o tipo de coleta de lixo é empregado.

Questão 5. Leia o seguinte post:

<https://discord.com/blog/why-discord-is-switching-from-go-to-rust>

Elenque e explique quais os trade-offs envolvidos.

Questão 6. A contagem de referências sozinha não é suficiente para garantir que todos os nós inacessíveis sejam liberados. Para isso é necessário tomar cuidados adicionais. Considere a representação da memória abaixo.



- (a)** Verifique a situação descrita fazendo o seguinte:
- (i) adicione os contadores de referência;
 - (ii) faça $p \leftarrow \text{NULL}$ e $q \leftarrow \text{NULL}$;
 - (iii) libere a memória dos nós com contadores zerados.
- (b)** Pesquise e sugira cuidados adicionais para evitar essa situação.