

Ciência da Computação
Estrutura de Dados I
Prof. André Kishimoto

Observação: As instruções sobre entrega, prazo e avaliação estão nas páginas 2-4 do documento.

Projeto Prático 1 (Proj1) - EM DUPLA

Para solucionar a P1, a dupla deve implementar o TAD Fila Dupla (Deque ou double-ended queue) estática, isto é, usando vetor / array de tamanho fixo.

Uma fila dupla ou deque (“deck”) permite inserir e remover elementos tanto no início quanto no fim da fila, combinando funcionalidades da pilha e da fila.

TAD Fila Dupla (Deque)

OPERAÇÃO	COMPORTAMENTO
InsertFront(deque, elem)	Inserir o elemento <i>elem</i> no início da fila dupla <i>deque</i> , se <i>deque</i> não estiver cheia. Pré-condição: A fila dupla <i>deque</i> é válida. Pós-condição: O início de <i>deque</i> contém o elemento <i>elem</i> OU erro se <i>deque</i> estiver cheia.
InsertBack(deque, elem)	Inserir o elemento <i>elem</i> no fim da fila dupla <i>deque</i> , se <i>deque</i> não estiver cheia. Pré-condição: A fila dupla <i>deque</i> é válida. Pós-condição: O fim de <i>deque</i> contém o elemento <i>elem</i> OU erro se <i>deque</i> estiver cheia.
RemoveFront(deque)	Remove e retorna o primeiro elemento da fila dupla <i>deque</i> , se <i>deque</i> não estiver vazia. Pré-condição: A fila dupla <i>deque</i> é válida. Pós-condição: Remove e retorna o primeiro elemento da <i>deque</i> OU erro se <i>deque</i> estiver vazia.
RemoveBack(deque)	Remove e retorna o último elemento da fila dupla <i>deque</i> , se <i>deque</i> não estiver vazia. Pré-condição: A fila dupla <i>deque</i> é válida. Pós-condição: Remove e retorna o último elemento da <i>deque</i> OU erro se <i>deque</i> estiver vazia.
Front(deque)	Retorna uma referência do elemento que está no começo da fila dupla (mas não o remove), se <i>deque</i> não estiver vazia. Pré-condição: A fila dupla <i>deque</i> é válida. Pós-condição: Retorna o primeiro elemento da <i>deque</i> sem removê-lo OU erro se <i>deque</i> estiver vazia.
Back(deque)	Retorna uma referência do elemento que está no fim da fila dupla (mas não o remove), se <i>deque</i> não estiver vazia.

	Pré-condição: A fila dupla <i>deque</i> é válida. Pós-condição: Retorna o último elemento da <i>deque</i> sem removê-lo OU erro se <i>deque</i> estiver vazia.
Create()	Cria e retorna uma fila dupla vazia. Pré-condição: N/A. Pós-condição: Uma nova fila dupla vazia é criada.
Size(deque)	Retorna a capacidade da fila dupla. Pré-condição: A fila dupla <i>deque</i> é válida. Pós-condição: N/A.
Count(deque)	Retorna a quantidade de elementos na fila dupla. Pré-condição: A fila dupla <i>deque</i> é válida. Pós-condição: N/A.
IsEmpty(deque)	Retorna true se a fila dupla estiver vazia ou false, caso contrário. Pré-condição: A fila dupla <i>deque</i> é válida. Pós-condição: N/A.
Clear(deque)	Esvazia a fila dupla (remove todos os elementos da fila). Pré-condição: A fila dupla <i>deque</i> é válida. Pós-condição: A fila dupla <i>deque</i> está vazia.

Desenvolvimento (6,0 pontos)

- Sua solução deve ser escrita apenas com a linguagem C++.
- Tente sempre trabalhar com arquivos .cpp/.h, modularizando o seu código.
- Além da implementação da fila dupla (deque), seu código deve conter um exemplo que mostre como usar a deque e que ela funciona corretamente.
- Os tipos de dados manipulados pela deque devem ser definidos por você.

Apresentação e explicação da solução (4,0 pontos)

- Além da sua solução escrita em C++, a dupla deverá gravar um vídeo de no máximo 5 (cinco) minutos, explicando como a fila dupla foi implementada e o seu funcionamento interno.
- A dupla deve se apresentar e aparecer no início do vídeo (algo bem simples e rápido como um “Oi, eu sou tal pessoa.” “E eu sou tal pessoa e vamos explicar como funciona um deque e como implementamos essa estrutura de dados.”).
- Para a explicação, você pode usar quaisquer recursos que te ajude, tais como animações, slides, captura do projeto rodando etc.
- Certifique-se que o vídeo esteja em qualidade alta e que todo o conteúdo seja legível (principalmente se aparecer código).

Identificação e referências

- Coloque sua identificação - nome e TIA - no início de cada arquivo de código, como comentário (use // no começo de cada linha que queira comentar).
- Inclua como comentário quaisquer referências (livros, artigos, sites, entre outros) usadas para solucionar o problema.

Entrega

- **Código:** Compacte todos os arquivos .cpp/.h ou o projeto completo criado na IDE que você está usando (mas sem os intermediários como bin e obj) no formato zip OU comite todos os arquivos .cpp/.h ou o projeto completo criado na IDE que você está usando (mas sem os intermediários como bin e obj) em um repositório git.
- **Vídeo:** O vídeo pode ser enviado para o Youtube e NÃO precisa ser público (pode ficar como não listado) OU, se o arquivo de vídeo tiver até 30MB, pode ser anexado no Moodle.
- **Arquivo texto (.txt):**
 - Se o código está em um repositório git, envie um arquivo txt no Moodle contendo sua identificação e o link do repositório.
 - Se o vídeo está no Youtube, envie um arquivo txt no Moodle contendo sua identificação e o link do vídeo no Youtube.
- **Prazo de entrega:** via link do Moodle até 16/04/2021 23:59.

Informações importantes sobre critérios de avaliação

Embora essa atividade seja uma avaliação da disciplina, sempre considero que as atividades também podem ser usadas para nos acostumarmos com o mercado de trabalho. Portanto, leve em consideração os seguintes critérios que vou aplicar na avaliação:

- Será descontado 1,0 (um) ponto caso a entrega não respeite o enunciado.
Exemplos:
 - O enunciado pede para enviar um arquivo compactado no formato zip, mas o arquivo enviado está no formato rar.
 - O enunciado pede um arquivo texto no formato txt, mas foi enviado um documento do Word.
 - Não há identificação nem referências (caso aplicável) nos arquivos de código.
 - O vídeo possui mais de 5 (cinco) minutos.
- Será descontado 1,0 (um) ponto caso o arquivo zip OU o repositório git contenha pastas e arquivos desnecessários.
Exemplo:
 - Pastas intermediárias criadas no processo de compilação (Debug, obj, bin, ...).
- O projeto deve ser desenvolvido em linguagem C++ e não em linguagem C. Caso a solução apresentada use funcionalidades da linguagem C e que tenham equivalentes em C++, será descontado 2,0 (dois) pontos.
Atente-se a esse detalhe quando estiver pesquisando e verificando exemplos na internet e outros materiais, principalmente de assuntos que não vimos até o momento (essa atividade pode ser resolvida só com o que foi visto em aula, com suas devidas adaptações).
Exemplo:
 - Declarar arrays de tamanho variável (padronizado no C99, mas erro em C++ pois não há suporte para VLA), ex. `int n = 10; char arr[n];`.

- Projeto que possui erros de compilação ou que trava durante a execução automaticamente perde 50% da nota máxima.

Sobre erros de compilação: considero apenas erros, não há problema se o projeto tiver warnings (apesar que warnings podem avisar possíveis travamentos em tempo de execução, como loop infinito, divisão por zero etc.).

Quando há necessidade de entrada de dados por parte do usuário, assumo que o usuário vai inserir as informações corretas (ex. tipos de dados corretos), a menos que o enunciado explicita que você deve garantir que os dados de entrada estejam corretos.

Em uma situação profissional, os itens indicados acima atrapalham (e muito) o trabalho da equipe. E o último item é gravíssimo (o ideal também é remover todos os warnings e sempre validar os dados).