TP 1 – Aeds 2

Nome: Ícaro Kened Torres Neto

# Introdução

O trabalho proposto consiste na criação de um algoritmo para solucionar a falta de exatidão na contagem média de alunos atendidos diariamente na cantina do Icex, UFMG. O algoritmo deve replicar com exatidão o processo que os alunos devem executar afim de comprar sua refeição na cantina. Processo esse que vai da fila do caixa, até a fila em que são servidos os alimentos, que consiste na fase final desta atividade.

O objetivo do algoritmo é, então , otimizar tal processo, fazendo uso das estruturas de dados ensinadas na disciplina de Aeds 2.

# Desenvolvimento

Para solucionar tal problema foram utilizadas estruturas de dados. Foram criados dois TADs, o primeiro com o objetivo de criar e inicializar as diversas filas necessárias no decorrer do algoritmo, e um outro TAD para criar pilhas. Para preencher a fila inicial do caixa, e as subsequentes, o TAD “FilaCant” foi de grande valia, uma vez que todas as funções necessárias para criação de filas estavam disponíveis no mesmo. O mesmo serve para o outro TAD citado acima, o “PilhaCant”, que contém todas as funções utilizadas no algoritmo para a criação de pilhas no algoritmo.

# Implementação

Estrutura de DadosPara as implementações do trabalho foram criados Tipos Abstratos de Dados Vetor, utilizando alocação dinâmica de memória, uma vez que não é possível mensurar o tamanho das filas que serão implementadas no algoritmo. Para a pilha, a alocação de memória estática foi utilizada, já que para esse tipo de estrutura de dados foi designado um tamanho máximo (30).

Funções

Os TADs criados apresentam as seguintes funções:

FFVazia(TFila \*Fila) : Faz com que a fila em questão fique vazia, o que foi utilizado no final do expediente.

Vazia(TFila Fila) :Verifica se a fila está vazia(não foi utilizada na implementação do algoritmo).

Enfileira(TItem x, TFila \*Fila) : Insere um item x no final da fila. Esta função foi de grande valia na implementação do algoritmo, nas iterações que visavam preencher as diversas filas apresentadas no problema.

Desenfileira(TFila \*Fila, TItem \*Item): Retorna o item x que está no início da fila, o retirando da mesma. Utilizado de forma conjunta com a função “Enfileira” acima, possibilitou o remanejamento dos alunos dentre as filas, inserindo – os em outra assim que fossem atendidos ou completassem seus objetivos.

FPVazia(TPilha \*Pilha): Faz a pilha ficar vazia.

Vazia(TPilha Pilha): Verifica se a pilha está vazia. Utilizado quando foi necessário verificar se existiam bandejas disponíveis para os alunos.

Empilha(TItem x, TPilha \*Pilha): Função que insere um item x no topo da fila.

Desempilha(TPilha \*Pilha, TItem \*Item): Função utilizada para retornara um item x no topo da filha, o retirando da mesma.

# Análise de Complexidade

As funções do algoritmo possuem a mesma complexidade, O(1).

# Resultados

No algoritmo podemos visualizar que uma média de 200 pessoas foram atendidas por implementação, numa média de tempo de 60.

# Conclusão

A primeira implementação foi organizada com um aumento de 10 bandejas na pila de bandejas a cada 12 segundos, como proposto no trabalho.

A segunda implementação faz com que 12 bandejas sejam adicionadas a cada 20 segundos. Sendo assim, podemos inferir que a primeira implementação obtém melhor desempenho, uma vez que o menor intervalo de tempo entre iteração na pilha compensa o menor número de bandejas adicionadas.