

UNIVERSIDADE FEDERAL DO PIAUÍ - CAMPUS SHNB

SISTEMAS DE INFORMAÇÃO

ALGORITMOS E PROGRAMAÇÃO II

Lucas Vinicius Silva dos Santos

Jederilson Sousa Luz

Erick McGregor Santos Lima

Trabalho Prático

1 - Introdução

O algoritmo proposto trata-se de um editor de texto simples, contendo apenas duas funções: A função **type** e a função **undo**. A função **type** trata-se de uma função para a entrada de dados, enquanto que a função **undo** serve para desfazer ações anteriores.

Cada função inserida possui um tempo que deverá ser inserido também no arquivo de entrada. Ex: *{type a, type b, undo 2} {1,3,5}*, o **type a** é inserido no **tempo 1**, o **type b** é inserido no **tempo 3** e **undo 2** é inserido no **tempo 5**.

Em relação ao funcionamento da função **undo**: cada undo deverá desfazer os comandos anteriores dentro do seu intervalo de tempo. Tomando o exemplo anterior, a função **undo 2** iria desfazer os comandos nos dois últimos segundos em relação ao seu tempo. No caso como seu tempo é 5, apenas a função **type b** no **tempo 3** seria desfeita por estar dentro do intervalo de ação do undo.

Ex:

Entrada:

{type a,type b,type c,undo 3},{1,2,3,5}

{type a,type b,undo 2,undo 2},{1,2,3,4}

{type a,undo 1,undo 1},{1,2,3}

{undo 1},{1}

{type a,type b,type c,undo 10},{1,2,3,1000}

Saída:

a

a

a

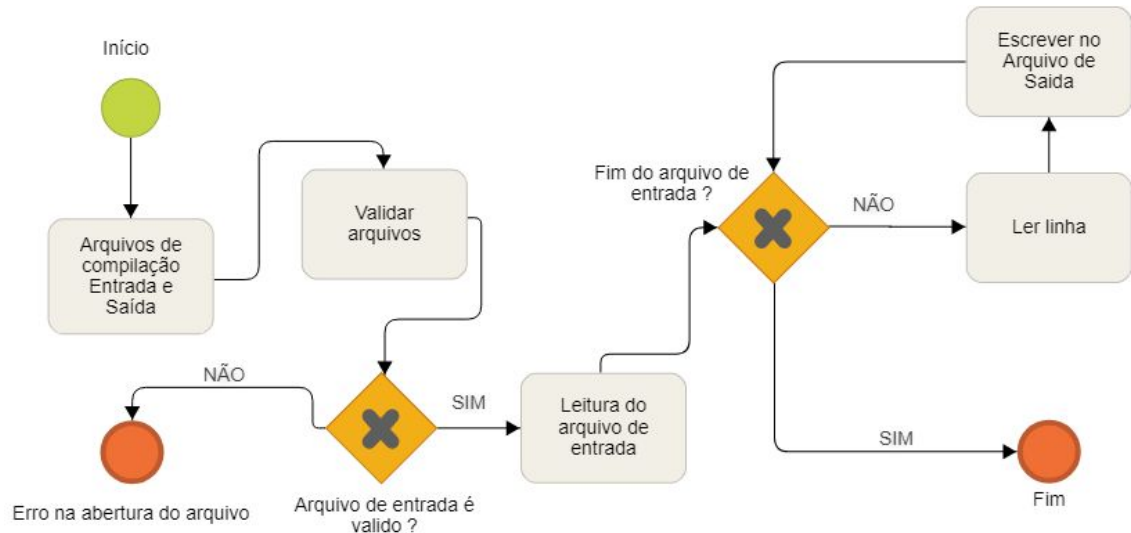
abc

2 - Solução Proposta

Para receber as entradas foi criado uma lista de *Registros* do tipo *Comandos* com os atributos **tempo**, **type** e **undo**. Percorrendo até o fim do arquivo (**EOF**), o programa recebe numa variável uma linha completa e processa a entrada alocando os dados de cada comando dentro da lista de *Comandos*, linha por linha.

A cada leitura completa de uma linha, o programa lê os comandos dessa linha da direita para a esquerda sequencialmente. Ao encontrar um **type C**, onde **C** é o caractere que acompanha o comando, **C** é escrito no arquivos de saída. Ao encontrar um comando **undo**, o programa calcula o tempo de **continuação (tempo T - undo N)**. Ex: $\{type\ a, undo\ 1\}, \{1, 2\}$, onde o **tempo T** = 2 e **undo N** = 1. Seguindo com a leitura, todos os comando seguintes com o **tempo T** maior ou igual ao valor de **continuação** são simplesmente ignorados.

O seguinte fluxograma representa o funcionamento o algoritmo, desde o seu início, suas condições e suas saídas:



Obs: Uma técnica de programação utilizada nesse algoritmo foi a busca sequencial, onde o algoritmo percorre as linhas do arquivo buscando e coletando os dados desejados.

3 - Análise de Desempenho

O programa se divide em duas seções principais: A leitura dos arquivos de entrada na função **recarregar** e a escrita no arquivos de saída com a função **escrever**.

A quantidade de comandos executados na função **recarregar** é linear à quantidade de linhas e comandos do arquivo de entrada segundo a fórmula $C = L * 2Co$, onde **C** é o custo, **L** é a **quantidade de linhas** e **Co** a **quantidades de comandos** total. O $2Co$ do cálculo se deve ao segundo loop de leitura dedicado ao **tempo** de cada comando, que se repete pela mesma quantidade de comandos.

Na função **escrever** há um loop para ler os comandos e outro para escrever os dados corretos no arquivo de saída, que irão ser executados pela mesma quantidade de linhas da entrada. Seu custo segue a fórmula $C = L * Co * Ca$, onde **Ca** é a **quantidade de caracteres escritos** na saída do programa e as demais variáveis iguais às da primeira fórmula.

4 - Conclusão

Por fim, o algoritmo de editor de texto, trata-se de uma aplicação para inserir e remover dados disponibilizados em um arquivo de entrada e exibir esses dados processados em um arquivo de saída. Para processar as entradas desse algoritmo, foi usada a técnica de busca sequencial para procurar e armazenar as informações pertinentes, tais como: valores dos **types**, os valores dos **undos** e os **tempos** que cada função possui para se obter a saída desejada. O programa determina quais informações **type X** deverão ser escritas no arquivo de saída, calculando a diferença (**tempo T - undo N**).

Um aspecto positivo desse algoritmo inclui um rápido tempo de execução, mesmo com arquivos de entrada grandes. Em contrapartida disso, um ponto negativo do algoritmo deve-se a forma que o mesmo trabalha, analisando todos os comandos mesmo que um **undo** fosse anulá-los, exigindo um maior processamento do computador. Uma solução mais otimizada poderia pular os comandos que não seriam considerados na saída.