CENTRO UNIVERSITÁRIO DE ANÁPOLIS – UniEVANGÉLICA
CURSO DE ENGENHARIA DE SOFTWARE

ALGORÍTIMO DE SISTEMA DE NOTAS ACADÊMICO

MATHEUS N. SAITO

ANÁPOLIS, GOÍAS
2023

Matheus N. Saito

ALGORÍTIMO DE SISTEMA DE NOTAS ACADÊMICO

Trabalho apresentado à disciplina de Analise e Complexidade de Algoritimos, no terceiro ciclo de aprendizagem.

Orientador(a): Prof. Wagner.

Anápolis, Goías
2023

Escolha das Estruturas de Dados
Com base nos requisitos, escolha as estruturas de dados mais adequadas para representar as informações relevantes na plataforma de educação digital. Pode ser necessário utilizar listas, árvores, grafos, entre outras estruturas, neste caso será utilizado uma Lista com Struct que utiliza arquivos para recuperar e gravar dados na linguagem C

Desenvolvimento de Algoritmos em Linguagem C
Desenvolvendo os algoritmos necessários para as operações requeridas pela plataforma. Isso pode incluir algoritmos de busca, ordenação, processamento de dados e outros, dependendo das funcionalidades específicas exigidas. Neste caso sera enviado um algoritmo de listagem.

Otimização de Código
Após a implementação inicial, o foco está na otimização do código para garantir eficiência e desempenho. Isso pode envolver a minimização do uso de recursos computacionais, a escolha eficiente de algoritmos e a aplicação de técnicas de otimização de código em nível de compilador, a complexidade deste algorítimo pode ser considerada O(n), sem uso de “for” ou estruturas de repetição, sendo um algorítimo simples e um processo cíclico via terminal.

Análise de Complexidade Computacional
Realizando uma análise detalhada da complexidade computacional dos algoritmos desenvolvidos. Classifique-os em P (solucionáveis em tempo polinomial), NP (não polinomial) ou PN-Completo (problema NP-difícil e, ao mesmo tempo, NP). Este algoritmo é de baixa complexidade, sendo problema solucionável e de baixo consumo computacional.

5. Testes e Validação
Testes extensivos para garantir que os algoritmos funcionem corretamente em diferentes cenários. Isso é crucial para garantir a confiabilidade da plataforma de educação digital, a única forma de “crashar” o algorítimo é o input de dados de tipos diferentes devido a ausência de validação e por erros de usabilidade que fechem o sistema.

6. Revisão e Iteração
Sempre é necessário fazer revisões regulares do código com a equipe envolvida no projeto. Se necessário, é preciso fazer iterações para aprimorar ainda mais a eficiência e corrigir possíveis problemas identificados durante os testes.

7. Implementação na Plataforma UniSoft
O algorítimo a ser entregue consome recursos de alocação de memoria, por ser uma Lista Encadeada, que utiliza uma “Struct” já pode ser considerado um algorítimo otimizado para plataforma UniSoft, garantindo que tudo funcione conforme esperado no ambiente real de uso.

A aplicação funciona via terminal, sendo reaproveitado da disciplina de Estrutura de Dados.
É possível conferir o algorítimo nesse link: GitHub