

Comentários sobre a Estruturação do Projeto

1. TAD Site

O Tipo Abstrato de Dados Site é organizado de forma que as informações de determinado site fiquem encapsuladas em uma única estrutura. Além das informações elementares (código, nome, relevância, link da página principal, quantidade de palavras chave e palavras chave), adicionamos o parâmetro booleano “ativo”, a fim de facilitar a verificação da atividade do site.

2. TAD Servidor

O Tipo Abstrato de Dados Servidor, responsável por manipular um conjunto de sites, foi implementado como uma lista sequencial dinâmica, tendo em sua estrutura um vetor que guarda TAM_MAX ponteiros para sites. Também foi incluído um inteiro, representando a quantidade de sites ativos.

```
struct servidor_{  
    SITE *sites[TAM_MAX];  
    int qtSitesAtivos;  
}
```

A escolha da lista sequencial se deve ao fato que, ao associar o código dos sites com o índice do vetor da lista, as consultas por sites acontecem de maneira eficiente, otimizando em grande parte o algoritmo.

Assim, foi adotada a seguinte estratégia: ao criar um servidor, são criados também os 10000 sites, dos quais a informação “ativo” vale falso, inicialmente. Dessa forma, as operações de consulta, inserção e remoção necessitam apenas verificar se um site está ativo ou não no servidor dado um código (que também é posição do site no servidor), o que permite que essas operações aconteçam em tempo constante $O(1)$. Por outro lado, se o servidor fosse implementado como uma lista encadeada dinâmica, haveria a necessidade de uma busca por todos os sites para realizar todas as operações supracitadas, às quais teriam complexidade linear $O(n)$.

É fato que a lista encadeada gastaria menos memória para os casos em que o servidor não estivesse com seu total de sites ativos, mas, como a quantidade de sites a serem armazenados é relativamente pequena, visto que os valores dos códigos são menores ou iguais a 10000, são ínfimos os riscos de overflow de memória, dada a arquitetura dos computadores atuais.

Portanto, visivelmente a escolha do tipo de lista e a lógica na qual o algoritmo se baseia, resulta em uma aplicação extremamente eficiente, em termos de tempo, para o problema proposto pela parte 1 do trabalho.