Leandro de Assis Madeira

Estrutura de Dados – Prof.ª Dr.ª Andreia Cristina Grisolio Machion

Fatec Ipiranga

Tarefa da Aula do dia 25/03/2021

**Algoritmo de Ordenação Utilizado:**

void insertion\_sort(int \* v, int n){

    int i, j, x;

    for(j=1; j<n; j++){

        x = v[j];

        for(i = j-1; i >= 0 && v[i] > x; i--){

            v[i+1] = v[i];

        }

        v[i+1] = x;

    }

}

**Algoritmos de Busca Utilizados:**

int busca\_sequencial (int x, int \*v, int n) {

    int i;

    for (i=0; i<n; i++) {

        cont++;

        if (v[i] == x) {

            return i;

        }

    }

    return -1;

}

int busca\_binaria (int x, int \* v, int n) {

    int ini=0, fim=n-1, meio;

    while (ini <= fim) {

        meio = (ini + fim) / 2;

        if (x == v[meio]) {

            return meio;

        }

        if (x > v[meio])

            ini = meio + 1;

        else

            fim = meio - 1;

    }

    return -1;

}

**Comparando os números do tempo de ordenação e das buscas:**



**Hardware utilizado:**

* Processador: Intel® Core™ i3-5005U CPU @ 2.00 GHz
* Memória Instalada (RAM): 4,00 GB

**Conclusão:**

Através da atividade, pôde-se perceber e compreender de forma mais clara o funcionamento dos algoritmos de busca. Foram abordados na atividade os algoritmos de busca sequencial e busca binária, combinado ao algoritmo de ordenação *Insertion Sort*. A principal proposta é executar os dois tipos de busca em um vetor que terá seu tamanho variado para a análise do desempenho dos algoritmos. De tal forma, foram realizadas duas comparações diferentes, a primeira utilizando o algoritmo de busca sequencial em um vetor não ordenado e comparando ao desempenho do algoritmo de busca binária (que necessita que o vetor esteja previamente ordenado), e a segunda utilizando o algoritmo de busca sequencial em um vetor também ordenado. Para tal experimento foram utilizados sempre termos iguais para a realização das buscas.

Através desses testes, os resultados obtidos demonstram que há diferenças notavelmente grandes entre o desempenho dos algoritmos nas condições estabelecidas. Ao executar o algoritmo de busca sequencial em um vetor não ordenado, o número de buscas realizadas é exorbitante, percorrendo todos os índices do vetor até encontrar o termo procurado, além de apresentar um desempenho impreciso e imprevisível devido à pseudoaleatoriedade dos termos gerados pelo vetor. Embora este primeiro método não necessite de tempo de execução para a ordenação do vetor, ele acaba realizando a busca de forma ineficaz, enquanto o algoritmo de busca binária realiza a busca executando um número ínfimo de buscas quando comparado ao algoritmo de busca sequencial com o vetor não ordenado. Executando o algoritmo de busca sequencial em um vetor ordenado, o número de buscas se reduz a, aproximadamente, 0,1% do número constatado ao executar o mesmo algoritmo em um vetor não ordenado, o que nos mostra uma extrema vantagem em se trabalhar com vetores ordenados, porém, esse número permanece alto quando comparado com o desempenho do algoritmo de busca binária.

Teoricamente, podemos dizer que o algoritmo de busca sequencial realiza comparações com todos os elementos do vetor até encontrar o termo procurado, enquanto o algoritmo de busca binária realiza uma operação que o leva diretamente ao meio do vetor. Essa operação é descrita pela formulação matemática , onde *m* representa o meio, *ini* representa o início do vetor () e *fim* representa o final do vetor (). Essa operação se repete, sempre reduzindo o tamanho do vetor pela metade, até chegar ao termo procurado entre os elementos pertencentes ao vetor. Matematicamente, podemos descrever o número de iterações realizadas como uma sequência . Logo, podemos descrever o número aproximado de iterações como .

De tal forma, foi possível observar através desse estudo que temos um desempenho muito mais eficaz na realização de buscas em vetores quando utilizamos o algoritmo de busca binária, mesmo necessitando de tempo de execução para a ordenação do vetor antes da realização da busca em si.

**Referências:**

FEOFILOFF, Paulo; Busca em Vetor Ordenado; Disponível em:

< [https://www.ime.usp.br/~pf/algoritmos/aulas/bubi.html](https://www.ime.usp.br/~pf/algoritmos/aulas/bubi.html%20) > Acesso: 30/03/2021.