EACH - Escola de Artes, Ciências e Humanidades
- Algoritmos e Estruturas de Dados - II - Sistemas de Informação

Prof. Helton Hideraldo Bíscaro

1º prova: 26/09/2007

1. Mostre que se um grafo G possui um circuito Euleriano, então todo vértice tem grau par.

Suponha que G é um grafo que tem um circuito Euleriano. Seja v um vértice qualquer de G.

O vértice v não pode ser isolado, pois o circuito Euleriano deve conter todos os vértices de G .O circuito Euleriano possui cada aresta de G incluindo todas as arestas incidentes a v. Considere um caminho que começa no meio de uma das arestas adjacentes ao início do circuito Euleriano e continua ao longo deste circuito e termina no mesmo ponto. Cada vez que o vértice v é visitado através de uma aresta de entrada, este vértice é "deixado" já que o caminho termina no meio de uma aresta. Já que cada circuito Euleriano passa em cada aresta de G exatamente uma única vez, cada aresta incidente a v é visitada uma única vez neste processo. Como o caminho que passa por v é feito através de arestas incidentes a v na forma de pares entrada/saída, o grau de v deve ser múltiplo de 2.

2 Escreva um algoritmo para verificar se, dados um grafo G e dois vértices i e j, existe um caminho ligando o vértice i ao vértice j. (O algoritmo não deve entrar em loop)

```
bool existeCaminho(int i,int j, TMatrix M)
{
    int n= M[0].lenght;
    int k,l;
    visitado[i]=true;
    for (k=0;k<n;k++)
        if((M[i][k]!=0)&&(visitado[k]==false))
        {
            if (k==j)
                return true;
            else
```

```
            return existeCaminho(k,j,M);

        }

    return false; }
```

3 Calcular o tempo para ordenar um arquivo de 80 Megabytes utilizando o algoritmo merge-sort. Use as seguintes informações para os cálculos:

   (a) Tempo médio de seek: 18ms
   (b) Tempo de latência rotacional: 8.3 ms
   (c) Taxa de transferência: 1,229 bytes/ms
   (d) Buffer de 20.000 bytes disponível para escrita
   (e) 4 Megabytes de memória Ram

Vamos criar 20 corridas de 4 Mega cada. Temos operações de leitura e escrita em 4 fases distintas:

1. Fase de Ordenação:
   (a) Leitura dos registros para a memória com objetivo de criar as corridas
   (b) Escrita das corridas ordenadas para o disco.
2. Fase de Intercalação
   (a) Leitura das corridas para a intercalação (Merge-Sort)
   (b) Escrita do arquivo final no Disco.

Criação das corridas: $20 \times (seek + latência\,rotacional) = 20 \times (18 + 8.3) = 526.0ms = 0.5s$

Tempo de transferência = $\frac{80000}{1.229 \times 1000} = 65.094s$

Tempo total de leitura e criação das corridas = $65.6s$

Tempo total de escrita das corridas no disco é idêntico, já que as operaões são as mesmas. $65.6s$

Tempo de Merge: Tem-se 20 corridas de 4 mega cada. Logo, cada buffer deve armazear $\frac{1}{20}$ de uma corrida e cada corrida necessita de 20 acessos para ser lida por completo. Isso implica num total de 400 seeks .

$400 \times (18 + 8.3) = 10520.0ms = 10.5s$. Mais 65 segundos para transferir 80 Mega como foi calculado anteriormente.

Total = $75.5s$

Escrita final do arquivo no disco: Precisaremos de 80000000/20000 = 4000 seeks (um seek a cada vez que buffer se enche) para a escrita.

$\frac{4000\times(18+8.3)}{1000}$: $105.2s$ mais o tempo de trensferência que ainda é de $65s$ .Total = $170.2s$

Tempo total= $65.6 + 65.6 + 75.5 + 170.2 = 376.9 = 6$ min e $17s$.

4 Explique a organização de discos em setores, clusters e extents. Suponha que queremos armazenar 20.000 registros de tamanho fixo num disco de 300 Mega, quantos cilindros são necessários para armazenar o arquivo supondo que cada registro ocupa 256 bytes e que o disco se organiza da seguinte forma:

(a) 512 bytes por setor

(b) 40 setores por trilha

(c) 11 trilhas por cilindro

(d) 1331 cilindros.

Setor é a menor porção endereçável de um disco. Um cluster consiste de um conjunto de setores logicamente contiguos (a contiguidade física depende do fator de intercalação). Extent é um conjunto de clusters consecutivos.

É possível armazenar dois registos por setor. Logo serão necessários 10.000 setores. O número de trilhas será de $\frac{10000}{40} = 250$.E finalmente o número de cilindros será de $\frac{250}{11} = 22.727$

5 Escreva um algoritmo para remover uma chave de uma tabela hash ordenada. (OBS: a ordenação deve ser preservada)

```
bool removeChave(int chave, TTable V)

{

    int i=h(chave);

    int tmp,j;



    while(V[i].chave>chave)

        i=rh(i)

    if(V[i].chave!=chave)

        return false;

    while(V[i].chave!=-1)
```

```
            {
                j=rh(i);
                V[i].chave=V[j].chave;
                i=j;
            }
        return true;
}
```