Universidade Federal da Bahia (UFBA)
Instituto de Computação (IME)
Departamento de Ciência da Computação (DCC)
MATA 54 - Estrutura de Dados e Algoritmos II – 2024.1 – Prof. George Lima

Estudo experimental de métodos de ordenação externa
05/04/2024

1 Objetivo, contexto e informações iniciais

Este trabalho tem como objetivo consolidar os conceitos sobre os métodos de ordenação externa vistos em sala. Características de seus desempenhos serão avaliadas experimentalmente. Deve-se considerar os métodos de ordenação balanceada multi-caminhos, ordenação polifásica e ordenação em cascata. Há a liberdade de incluir variações desses ou outros métodos, mas pelo menos os três citados devem fazer parte do trabalho. Há material de referência que explora em profundidade esses e outros métodos de ordenação [1]. Consultas a esta e outras referências complementares podem ajudar [2, 3, 4].

O trabalho deverá ser entregue em forma de relatório. O código fonte correspondente à implementação deverá ser disponibilizado. O relatório deve conter explicações claras e concisas sobre os seguintes itens:

- Descrição dos métodos estudados.
- Descrição dos experimentos e implementação dos métodos. O código fonte deve estar disponível e compartilhado através de algum portal a partir do qual pode-se compilá-lo e executá-lo de forma remota. Informações para permitir o compartilhamento devem estar presentes de forma clara no relatório. Se for necessário, usar o endereço eletrônico `gmalima@gmail.com` para permitir acesso ao compartilhamento (não enviar o trabalho para este e-mail).
- Apresentação dos resultados experimentais e discussão sobre os aspectos relevantes observados nos experimentos.
- Considerações finais sobre o trabalho, ressaltando os aspectos mais importantes.
- Todas as fontes bibliográficas utilizadas deverão ser citadas de forma apropriada.

O trabalho pode ser feito em equipe de no máximo três componentes e será avaliado em função do relatório e da implementação. Cada um destes aspectos com pesos iguais. Clareza, correção e precisão das informações contidas no texto do relatório têm peso importante na avaliação. A implementação será avaliada executando o código para exemplos conhecidos. Serão atribuídas nota zero a quaisquer trabalhos com indícios de plágio.

O relatório deverá ser entregue até dia 28/05/2024 e as composições das equipes devem ser informadas até dia 14/04/2024. Tanto o relatório quanto informações sobre a formação de equipes devem ser entregues em local indicado na página da disciplina no Ava Moodle. Após 14/04/2024 não será possível modificar as composições das equipes.

2 Avaliação experimental e apresentação dos resultados

Os métodos de ordenação externa estudados devem ser comparados experimentalmente. Os valores a serem ordenados serão numéricos e gerados aleatoriamente. Como métrica de comparação, deve-se usar a taxa de processamento média em função do número de sequências inicial geradas, como explicado a seguir. Além disso, objetiva-se investigar como o número de sequências iniciais varia em função do tamanho da memória interna.

2.1 Métricas de avaliação

Suponha um arquivo a ser ordenado com n registros e assuma que a memória principal só possui capacidade para m registros e que o sistema só tem capacidade de manter abertos concomitantemente k arquivos. Para todos os métodos, gera-se, antes da fase de intercalação, um número inicial r de sequências ordenadas (*runs*). Para tanto, usa-se o método visto em sala conhecido como seleção natural, que faz uso de uma *heap* mínima (assume-se aqui ordem ascendente de chaves dos registros). Os registros a serem ordenados serão representados por valores inteiros, que devem ser gerados aleatoriamente durante a fase de geração das sequências iniciais.

Para medir o esforço que cada algoritmo faz para intercalar as sequências iniciais, define-se a taxa de processamento $\alpha(r)$, calculada da seguinte forma:

$$\alpha(r) = \frac{\text{número total de operações de escrita sobre os registros do arquivo}}{\text{número total de registros}}.$$

Desta forma, para uma sequência arbitrária de registros e k arquivos abertos, gera-se as r sequências iniciais. Note que o número registros a serem considerados para valores fixos de r e k não é constante. Considere os valores de $k = 4, 6, \ldots, 12$ e $m = 3, 15, \ldots, 60$. Os valores de r serão aqueles contidos no conjunto $R = \{i \times j \leq 5000 | i = 1, 2, \ldots, 10; j = 10, 20, \ldots, 1000\}$. O valor de n é aquele que foi suficiente para gerar as r sequências iniciais.

Recomenda-se que o cálculo de $\alpha(r)$ considere ao menos 10 repetições para cada valor de r e k. Outra observação relevante está relacionada à geração dos valores a serem ordenados. Como o valor de n não será determinado de antemão, recomenda-se que valores numéricos aleatórios sejam gerados até que as r sequências iniciais sejam construídas. Por fim, observar que a disposição das sequências iniciais deverá estar de acordo com cada uma dos algoritmos de ordenação. Por exemplo, na ordenação balanceada, deve-se usar apenas $k/2$ dos arquivos para conter as sequências iniciais. Para os outros dois métodos de ordenação, estas sequências podem estar contidas em $k - 1$ arquivos.

Em cada um dos experimentos solicita-se ainda que seja avaliado o tamanho médio das sequências. Sabe-se que o tamanho das sequências iniciais depende de m, pois quanto maior a memória interna, maiores tendem a ser tanto as sequências iniciais quanto aquelas geradas durante o processo de intercalação. Para avaliar o tamanho das sequências iniciais e como estas crescem ao longo da ordenação, considere que há r_j sequências na fase j de ordenação, com $r_0 = r$ representando as sequências iniciais. Defina o fator $\beta(m, j)$ da seguinte forma

$$\beta(m, j) = \frac{1}{m\ r_j} \sum_{i=1}^{r_j} |S_{i,j}^m|,$$

com $|S_{i,j}^m|$ representando o tamanho da i-ésima sequência gerada na fase j da ordenação quando a capacidade da memória é de m registros.

Como pode ser observado, $\alpha(r)$ mede o esforço do método de ordenação no processo de intercalação, sendo, portanto, uma métrica natural de comparação dos diferentes métodos de ordenação. O fator $\beta(m, j)$, por sua vez, visa a avaliar a efetividade do método de geração de sequências iniciais (para $j = 0$) e como o tamanho médio das sequências evoluem durante a ordenação (para $j > 0$).

2.2 Resultados experimentais

Os resultados para $\alpha(r)$ deverão ser apresentados em forma de gráficos de linhas. No eixo das abscisas estarão representadas as quantidades r de sequências iniciais geradas. No eixo das ordenadas estarão os valores de $\alpha(r)$ correspondentes. Haverá um gráfico para cada método de ordenação. Em cada gráfico, o comportamento de $\alpha(r)$ para cada valor de k deverá ser exibido. Procure manter os diferentes gráficos na mesma escala para facilitar a comparação.

Com relação ao fator $\beta(m, j)$, é suficiente indicar seu comportamento para os vários valores de m considerados apenas para $j = 0$. O objetivo aqui é verificar como o valor de m influencia a geração das sequências iniciais. Desta forma, um gráfico de linha onde o eixo das abcissas representa m e o das ordenadas contém os respectivos valores de $\beta(m, 0)$ é suficiente.

3 Especificação da entrada e saída para correção da implementação

Como mencionado, a correção da implementação usará exemplos (geralmente pequenos) para os quais a resposta esperada é conhecida. Portanto, casos de testes serão submetidos de forma a verificar se: (a) as sequências iniciais estão sendo geradas de forma consistente; (b) os algoritmos de ordenação estão processando as sequências corretamente e no número de passos esperados. Os casos de testes serão compostos de uma sequência de n valores inteiros, dados como entrada. A saída conterá informação suficiente para que seja possível acompanhar o processo de ordenação e verificar a correção dos cálculos de $\alpha(r)$ e $\beta(m, j)$. A seguir as especificações da entrada e saída serão dadas. Os três métodos a serem avaliados serão representados pelas letras `B` (ordenação balanceada multi-caminhos), `P` (ordenação polifásica) e `C` (ordenação em cascata).

3.1 Entrada

A entrada é dada em três linhas. A primeira conterá o método a ser considerado (`B`, `P` ou `C`). A segunda linha fornece os valores de m, k e r, nesta ordem. A terceira linha conterá n valores inteiros a serem considerados para a ordenação. Por exemplo, para $m = 3$, $k = 4$, $r = 3$ e $n = 17$, um exemplo de entrada para o método de ordenação balanceada poderia ser dada por:

```
B
3 4 3 17
7 1 5 6 3 8 2 10 4 9 1 3 7 4 1 2 3
```

3.2 Saída

A saída é composta de várias linhas, que indicarão as fases de ordenação, os valores de $\beta(m, j)$ correspondentes a cada fase ao longo da ordenação, as sequências sendo ordenadas e, por fim, o valor encontrado para $\alpha(r)$. Para a entrada fornecida anteriormente a saída seria:

```
fase 0 1.56
1: {1 5 6 7 8}{1 3 4 7}
2: {2 3 4 9 10}
fase 1 2.33
3: {1 2 3 4 5 6 7 8 9 10}
4: {1 3 4 7}
fase 2 4.67
1: {1 1 2 3 3 4 4 5 6 7 7 8 9 10}
final 2.00
```

Note que antes de se executar a fase $j + 1$, o valor de $\beta(m, j)$ e a disposição das sequências na fase j são indicados. No exemplo fornecido, $\beta(m, 0) = 1.56$ e há três sequências iniciais, duas gravadas no arquivo 1 e a outra no arquivo 2. Os arquivos vazios não precisam ser informados. Após se executar a fase 1 da ordenação, duas sequências são geradas, gravadas nos arquivos 3 e 4. Como o tamanho médio destas sequências é $14/2 = 7$, o valor de $\beta(m, 1) = 2.33$. Por fim, na fase $j = 2$, estas duas sequências são intercaladas, produzindo a sequência final com tamanho 14, portanto $\beta(m, 2) = 14/3 = 4.67$. Na última linha da saída o valor de $\alpha(r)$ é fornecido. No exemplo, $\alpha(3) = 2$, pois o número de registros processados é $5 + 5 + 4 + 10 + 4 = 28$.

Pelo exemplo apresentado é possível notar que apenas $n = 14$ valores, dos 17 fornecidos, foram utilizados para construir as $r = 3$ sequências iniciais. Quando as r sequências iniciais forem formadas, antes de se formar a sequência $r+1$, os demais valores são descartados e não farão parte da ordenação.

Usando o mesmo exemplo da entrada, mas considerando o método de ordenação `P` (ordenação polifásica), a saída seria:

```
fase 0 1.56
1: {1 5 6 7 8}
2: {1 3 4 7}
```

```
3: {2 3 4 9 10}
fase 1 4.67
4: {1 1 2 3 3 4 4 5 6 7 7 8 9 10}
final 1.00
```

Referências

[1] Donald D. Knuth. *The Art of Computer Programming: Sorting and Searching.* Addison-Wesley, 3rd edition, 1973.

[2] Robert Sedgewick. *Algoritms in C++.* Addison-Wesley, 1st edition, 1992.

[3] Alan Tharp. *File Organization and Processing.* John Wiley & Sons, Inc, 1988.

[4] Nivio Ziviani. *Projeto de Algoritmos com implementações em PASCAL e C.* Addison-Wesley, 2nd edition, 1992.