

Trabalho do 3º bimestre

1 Introdução

Este trabalho tem como objetivo a realização de uma análise experimental de algoritmos de ordenação.

O trabalho é em equipe. O compartilhamento de informações entre as equipes é permitido (e aconselhado), mas o compartilhamento de código não é permitido. Trabalhos que tenham código igual serão anulados. Veja a Resolução N° 008/2007-COU para as possíveis sanções disciplinares.

Equipe: até duas pessoas.

Data de entrega: até o dia 17/10/2011 às 23:00h

Forma de entrega: o código do trabalho deve ser enviado para o email `malbarbo@gmail.com`, em um arquivo compactado com o nome `codigodisciplina_turma_ra1_ra2.zip` (substitua `codigodisciplina` pelo código da disciplina, `turma` pelo número da turma e `ra1` e `ra2` pelo número dos RA's dos membros da equipe). O formato de compactação **deve ser zip**, outros formatos não serão aceitos. Tenha o cuidado de **não enviar arquivos compilados**. O relatório escrito descrevendo os resultados dos experimentos deve ser entregue para o professor na sala de aula.

2 Descrição

Realizar experimentos práticos e indicar (baseado no resultado dos experimentos) qual é o algoritmo de ordenação mais adequado para cada caso. São dados vários casos:

1. Números inteiros de 32 bits
2. Números inteiros de 16 bits
3. Números reais distribuídos uniformemente no intervalo $[0, 1)$

Os algoritmos candidatos são: heapsort, quicksort, ordenação por contagem e ordenação por balde.

3 Experimento

Para cada caso, você deve determinar quais algoritmos podem ser utilizados, e realizar o seguinte experimento:

- Executar cada algoritmo e calcular o seu tempo de execução para sequências com 100000, 200000, 300000, ..., 1000000 elementos. Cada sequência deve ser gerada aleatoriamente. O tempo de execução é calculado pela média dos tempos de 10 execuções para a mesma entrada, descartando o melhor e o pior tempo. Tenha o cuidado de medir apenas o tempo de execução da função de ordenação, não mossa o tempo necessário para gerar os dados.

- Confirme que o seu algoritmo esta funcionando corretamente, checando (com uma função) se a sequência resultante está ordenada.

O programa que realizará os experimentos pode ser escrito em C, C++, pascal ou python. O programa deve poder ser compilado e executado em um sistema Linux. Além do código do seu programa, você deve fornecer um script executável chamado `executar`, que quando for executado compile e execute o seu programa.

Quando seu programa for executado ele deve realizar os experimentos e escrever na tela os resultados de cada experimento em forma de tabela, como no exemplo a seguir:

Caso 1

| | alg1 | ... | alg2 |
|---------|------|-----|------|
| 100000 | 10 | | 12 |
| 200000 | 20 | | 21 |
| ... | | ... | |
| 1000000 | 100 | | 98 |

Caso 2

...

Caso 3

...

Neste resultado, considerando o caso 1 e a sequência com 100000 elementos, o algoritmo alg1 demorou 10ms e o alg2 demorou 12ms.

4 Relatório

Além do código do programa, cada equipe deve entregar um relatório dos experimentos realizados. O relatório deve conter no mínimo:

- Configuração do sistema onde os experimentos foram executados (processador, memória, sistema operacional, compilador, opções de compilação, etc)
- Descrição do processo de geração dos dados de entrada
- Quais algoritmos foram escolhidos para serem executados para cada caso e a justificativa da escolha
- Tabela e gráfico (conforme exemplo a seguir) com os resultados de cada experimento
- Análise dos resultados e a indicação do algoritmo mais adequado para cada caso
- Dificuldades encontradas
- Conclusões finais

Os gráficos devem conter as mesmas informações das tabelas. O eixo X corresponde ao tamanho da sequência e o eixo y o tempo de execução. Veja o exemplo a seguir.

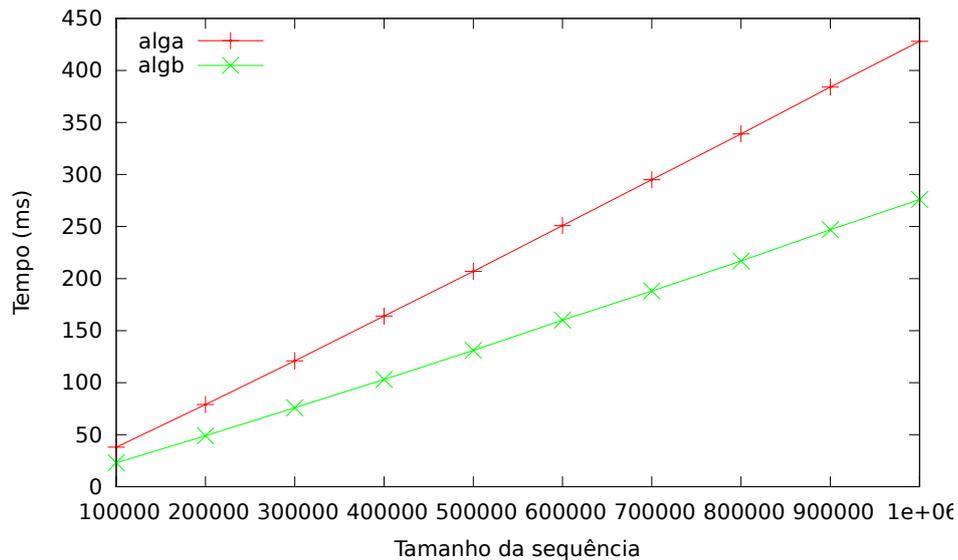


Figura 1: Tempo de execução dos algoritmos para o caso 1

5 Critérios de avaliação

Este trabalho vale 3,0 pontos na média do bimestre (isto mesmo, a prova vale 8,0 e este trabalho 3,0, além do desafio, que vale mais 1,0, totalizando 12,0).

Este trabalho será avaliado segundo os critérios:

- Clareza, organização e completude do relatório
- Clareza, organização e completude do código

6 Desafio

Este desafio vale 1,0 extra na nota do bimestre.

O desafio consiste em resolver o problema 8.4-4 do livro do Cormen. Para confirmar que a sua solução é mais adequada do que a aplicação direta da ordenação por balde, você deve apresentar um gráfico (como descrito anteriormente) dos dois algoritmos: ordenação por balde sem modificação e ordenação por balde com a modificação para o problema. Observe que o algoritmo modificado deve apresentar melhor tempo de execução. Tenha o cuidado de gerar os dados de entrada respeitando a descrição do problema. Se você quiser ter certeza que os dados estão sendo gerados de acordo com a descrição, venha conversar com o professor.

A resolução deste desafio deve ser entregue junto com o trabalho (mesmo código e mesmo relatório).