

Projeto e Análise de Algoritmos
2ª avaliação

1. Datas e Notas

Este trabalho vale 80 (oitenta) pontos e vale para a 2ª avaliação da disciplina. O trabalho poderá ser realizado em equipes de até 3 alunos.

Os problemas foram sorteados em sala e a distribuição ficou como segue:

| Equipe | Problema |
|---------------------------|------------------------------|
| João, Luiz e Elixandre | Segmento de Soma Máxima |
| Carina, Mateus e Matheus | Conjunto Independente Máximo |
| Medina, Renan | Par Mais Próximo |
| Bruno, Vilson e Emerson | Correspondência de Strings |
| Larissa, Leonardo e Vitor | Multiplicação de Matrizes |
| Marcelo | Multiplicação de Polinômios |

Para cada problemas, a equipe deve escolher pelo menos duas soluções para o problema sorteado (as soluções estão listadas na seção 4 deste documento. Esta escolha deve ser feita e enviada até o dia **21/06/2018** via e-mail: adri.postal@gmail.com ou diretamente com a professora. Os alunos que não cumprirem este prazo irão receber nota 0 (zero) nesta parte da avaliação.

O trabalho completo deve ser entregue até dia **17/07/2018**, até **12hs**, via Moodle, conforme descrito na seção 3.

2. O que deve ser feito

O trabalho consistirá de 5 partes:

1. Descrição completa do problema e das soluções escolhidos pela equipe;
2. Análise matemática de cada solução;
3. Implementação de **todas** as soluções escolhidas pela equipe;
4. Análise empírica de cada solução;
5. Apresentação do trabalho em forma de seminário.

A definição de cada parte está definida nas próximas seções (2.1 a 2.5).

2.1. Descrição do problema e das soluções

- Descrição sobre o problema escolhido: o problema, suas entradas, a saída esperada e qualquer outra informação relevante sobre o problema. Não esqueçam de incluir referências bibliográficas.
- Descrição de cada uma das soluções: as soluções que DEVEM ser trabalhadas estão indicadas na seção 4. Esta descrição deve conter a ideia geral da solução, a técnica de projeto envolvida no desenvolvimento da solução, o pseudocódigo (baseado no modelo adotado na disciplina) da solução e qualquer outra informação relevante sobre o problema. Não esqueçam de incluir referências bibliográficas.

2.2. Análise matemática

Neste item, deve ser analisado o pseudocódigo de cada solução indicada, conforme visto em sala de aula no capítulo 3.

No documento a ser gerado para este item deve estar descrito:

1. Operação básica de cada algoritmo (ou operações);
2. A análise de cada algoritmo (utilizando somatórios ou recorrências);
3. A análise realizada depende apenas do número de elementos? Se não, analisar melhor, médio e pior casos.

Ao final da análise matemática, após resolver os somatórios ou recorrências encontradas durante a análise, deve-se indicar a qual classe de eficiência aquele algoritmo pertence.

2.3. Implementação

Cada equipe deve escolher uma linguagem de programação (C, C++ ou Java) e implementar TODAS as soluções escolhidas para o problema escolhido. O código deve ser bem documentado, com comentários pertinentes. A execução dos testes para a correção do trabalho será realizada via console, em máquina com sistema Unix. Não será realizada execução via ambiente de edição (Netbeans, Eclipse, QT, etc). Não usar bibliotecas para Windows.

O formato da entrada e da saída devem ser descritos no documento a ser entregue, além de serem incluídos, junto ao código fonte, os conjuntos de entrada utilizados para a análise. Se a análise possuir melhor, médio e pior casos, deverá ser indicado os conjuntos utilizados para essas análises.

É neste item, também, que devem ser definidas as estruturas de dados utilizadas para representar a solução, se foi necessário espaço extra, etc.

2.4. Análise Empírica

Após a realização da implementação, ela deve ser analisada de forma empírica, conforme visto em sala de aula no capítulo 3.

No documento a ser gerado para este item deve estar descrito:

1. A métrica de eficiência que será medida e a unidade desta métrica: neste trabalho **não deve-se usar APENAS o tempo de execução**. A métrica *será* o número de vezes que

a operação básica (ou operações básicas) é executada. Vocês podem incluir o tempo de execução juntamente com o número de operações;

2. Tipos de dados que cada solução utiliza;
3. As características da amostra de entrada (faixa, tamanho, etc) de cada solução;
4. A análise realizada depende apenas do número de elementos? Se não, analisar melhor, médio e pior casos;

A análise dos dados obtidos deve ser mostrada através de tabelas e gráficos, além de indicar a qual classe de eficiência o algoritmo analisado pertence. **Não utilizar gráficos de colunas na análise:** esse tipo de gráfico não mostra o crescimento assintótico das funções.

Também deve ser realizado um comparativo entre a análise matemática e a empírica de cada algoritmo implementado, mostrando suas conclusões sobre a diferença (ou não) encontrada entre as duas análises e o que isso significa na prática.

Por fim, deve-se realizar um comparativo entre as soluções diferentes para o problema escolhido: qual é a melhor solução? Por que? A equipe deve justificar todas as suas conclusões, baseados nas análises obtidas durante o trabalho (não baseado em trabalhos encontrados na Internet).

2.5. Apresentação de seminários

Cada equipe deverá apresentar um seminário de 20 minutos sobre o trabalho realizado. As apresentações serão no dia 19/07/2018, a partir das 15hs, na sala de aula. A ordem de apresentação é a mesma mostrada na tabela de sorteio.

A apresentação deve conter, no mínimo:

- Problema resolvido;
- Soluções implementadas;
- Conclusões sobre as análises.

3. Documento a ser entregue

Deve ser entregue um documento digitado, no formato do artigo do EPAC (modelo em .doc e .tex em anexo a esta descrição).

A organização deve-se dar em, pelo menos, 7 seções (não há limite mínimo e máximo de páginas, o documento deve ser completo):

1. Introdução;
2. Descrição do problema escolhido (incluindo cada uma das soluções que serão abordadas, conforme descrito na seção 2.1 deste documento);
3. Soluções desenvolvidas (conforme descrito na seção 2.3 deste documento);
4. Análises matemáticas (conforme descrito na seção 2.2 deste documento, incluindo a comparação entre cada solução encontrada);
5. Análises empíricas (conforme descrito na seção 2.4 deste documento, incluindo a comparação entre cada solução encontrada);
6. Conclusões/Considerações Finais sobre o trabalho, incluindo as comparações entre as soluções encontradas e o comparativo com a justificativa de qual a melhor solução;

7. Referências bibliográficas (formato do TCC).

Todo o trabalho deve ser entregue via Moodle, até a data e horário indicados na seção 1 desta descrição. Cada algoritmo deve estar organizado em uma pasta separada, com todos os arquivos utilizados para sua implementação (códigos fontes, arquivos de teste, etc) e análise. Todas as pastas devem ser compactadas em um único arquivo, juntamente com o relatório técnico, com o nome do problema escolhido para solução (`ProblemaTal.zip`). Todos os códigos fontes devem estar identificados com o nome do da equipe. Basta que apenas um dos integrantes submeta o trabalho.

4. Opções de Problemas

Problema e suas soluções a serem implementadas (cada equipe deve escolher um problema e duas soluções):

1. Multiplicação de Matrizes:
 - a) Força Bruta
 - b) Divisão e Conquista (Strassen)
2. Par Mais Próximo:
 - a) Força Bruta
 - b) Divisão e Conquista
3. Casco Convexo (*Convex-Hull*):
 - a) Força Bruta
 - b) Divisão e Conquista
4. Problema da atribuição:
 - a) Força Bruta
 - b) Algoritmos Gulosos
5. Caixeiro Viajante:
 - a) Força Bruta
 - b) Algoritmo de Christofide
6. Problema da mochila:
 - a) Programação Dinâmica
 - b) Algoritmos Gulosos
7. Correspondência de Strings:
 - a) Algoritmo de Rabin-Karp
 - b) Algoritmo de Horspool
 - c) Algoritmo de Boyer-Moore
 - d) Correspondência de strings com AFs
 - e) Algoritmo de Knuth-Morris-Pratt
8. Árvore Geradora Mínima:
 - a) Algoritmo de Kruskal
 - b) Algoritmo de Prim
 - c) Algoritmo de Boruvka

9. Caminho Mínimo em Grafos:
- a) Algoritmo de Bellman-Ford
 - b) Algoritmo de Johnson
 - c) Algoritmo de Dijkstra
 - d) Algoritmo de Floyd-Warshall
10. Multiplicação de Grandes Inteiros:
- a) Força Bruta
 - b) Multiplicação à la Russe
 - c) Karatsuba
11. Conjunto Independente Máximo:
- a) Algoritmo Naive
 - b) Algoritmo de Ramsey
 - c) Algoritmo Bron-Kerbosch
 - d) Algoritmo Clique-Removal
12. Multiplicação de Polinômios:
- a) Força Bruta
 - b) Divisão e Conquista
13. Busca em Grafos:
- a) Profundidade
 - b) Largura
14. Segmento de Soma Máxima¹:
- a) Força Bruta
 - b) Divisão e Conquista
 - c) Programação Dinâmica

¹ Problema inspirado no livro: J.L.Bentley. *Programming Pearls*. ACM Press, 2nd edition, 2000.