

Aula Inaugural



Ciência da Computação
Universidade Federal de São Carlos

Programação e Estruturas de Dados

Tiemi Christine Sakata (tiemi@ufscar.br)

Disciplina do Mestrado

- No mestrado, as disciplinas representam a parte mais simples
- Importância da disciplina para o mestrado \Rightarrow desligamento do aluno regular ou não aceite em próximas disciplinas / mestrado
- Exige tempo e decisão
- Espera-se que o aluno seja capaz de estudar em profundidade. Não tem listas resolvidas, ou apostilas. Procure livros.

- Espera-se seriedade. Plágio é inaceitável.
- Cancelamento de disciplina \Rightarrow ir até a secretaria e cancelar a disciplina. Senão, o aluno fica com E na disciplina prejudicando seu histórico (prazos no site do PPGCCS)
- Prazos \Rightarrow 6 disciplinas em 18 meses e 3 anos para co-validar as disciplinas cursadas. Exame de qualificação e teste de idioma em 12 meses e defesa em 24 meses.
- Ensino público \neq gratuito

Mestrado é pesquisa

- Muita leitura em inglês
- Revisão bibliográfica no tema da pesquisa
- Leitura para compreensão **aprofundada** de artigos científicos
- Escrita de tipos diferentes de texto sobre o que você leu de outros autores e sobre seu próprio trabalho: Projeto de pesquisa, Qualificação, Relatório(s), Artigo(s) científico(s), Dissertação
- Programação (na maioria dos casos)
- Experimentação, coleta de resultados, análise crítica desses resultados
- Apresentação do seu trabalho em congresso (provavelmente)

Mestrado

- Qual a sua prioridade para o mestrado neste momento?
- Você vai precisar de muito mais tempo do que exigido pelas aulas.
- Você vai ter que se organizar sozinho.
- Tenha um orientador:
 - Ele tem muitas ocupações
 - Ele tem mais experiência de pesquisa do que você
 - Ele não fará o trabalho por você
- Aluno especial não é aluno de mestrado!

Algoritmos e Estruturas de Dados

“Um algoritmo é uma sequência não ambígua de instruções que é executada até que determinada condição se verifique.” (Wikipédia)

Algoritmos manipulam dados e esses dados devem estar organizados de forma coerente (estrutura de dados).

Uma estrutura de dados é um modo de armazenar e organizar dados com o objetivo de facilitar o acesso e modificações. Nenhuma ED única funciona bem para todas as finalidades e, por isso, é importante conhecer os pontos fortes e limitações de várias delas.

Os algoritmos fazem parte do dia-a-dia. O conceito de algoritmo é frequentemente ilustrado pelo exemplo de uma receita, embora muitos algoritmos sejam mais complexos. Eles podem repetir passos (fazer iterações) ou necessitar de decisões (tais como comparações ou lógica) até que a tarefa seja completada. Um algoritmo corretamente executado não irá resolver um problema se estiver implementado incorretamente ou se não for apropriado ao problema.

Sabe-se que algoritmos manipulam dados. Quando estes dados estão organizados (dispostos) de forma coerente, caracterizam uma forma, uma estrutura de dados. São a organização e os métodos que manipulam esta determinada estrutura que lhes conferem singularidade. A escolha de uma estrutura de dados apropriada pode tornar um problema complicado em um de solução bastante trivial. O estudo das estruturas de dados está em constante desenvolvimento (assim como o de algoritmos), mas, apesar disso, existem certas estruturas clássicas que se comportam como padrões.

Algoritmos e Estruturas de Dados

Para compreender uma estrutura de dados:

1. como os dados são armazenados
2. os algoritmos para manipular os dados
3. as características de desempenho das estruturas de dados para melhor escolha

Estruturas de dados e algoritmos estão intimamente ligados. Não se pode estudar estruturas de dados sem considerar os algoritmos associados a elas, assim como a escolha dos algoritmos em geral depende da representação e da estrutura dos dados.

Para compreender uma estrutura de dados é necessário: primeiro, saber como os dados são armazenados na memória do computador. Segundo, deve estar familiarizado com os algoritmos que manipulam a informação contida em uma estrutura de dados. Terceiro, deve conhecer as características de desempenho das estruturas de dados para, se necessário, selecionar dentre elas a que mais se adequa a uma determinada aplicação.

Eficiência

- Supondo computadores infinitamente rápidos e memória gratuita, você teria alguma razão para estudar algoritmos?
- Algoritmos diferentes que resolvem o mesmo problema podem ser diferentes em termos de eficiência. Isso pode ser mais significativo do que diferenças de hardware e software.
- Com capacidade cada vez maior dos computadores, resolvemos problemas cada vez maiores.
- É nos problemas maiores que as diferenças entre a eficiência dos algoritmos se tornam notáveis.

Se os computadores fossem infinitamente rápidos, qualquer método correto para resolver o problema serviria. Na maior parte das vezes, você utilizaria o método mais fácil de implementar. Mas a realidade é que os recursos devem ser usados com sensatez, e algoritmos eficientes em termos de tempo ou espaço são necessários.

Uma sólida base de conhecimento e técnica de algoritmos é uma das características que separam os programadores verdadeiramente qualificados dos novatos. Com a moderna tecnologia de computação, você pode executar algumas tarefas sem saber muito sobre algoritmos; porém, com uma boa base em algoritmos, é possível fazer muito, muito mais.

Objetivos

- definir e diferenciar as diversas estruturas de dados;
- manipular estruturas de dados utilizando algoritmos;
- selecionar e construir estruturas de dados adequadas para aplicações específicas;
- comparar estruturas de dados por meio de adequação ao problema.

Ementa

- Recursividade.
- Notação Assintótica e Introdução à Análise de Algoritmos.
- Pesquisa e ordenação em memória primária.
- Tipos abstratos de dados, arranjos e listas.
- Filas e pilhas.
- Árvores
- Grafos.
- Hashing.

Bibliografia

- T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein. Algoritmos - Teoria e Prática. 3ed. Campus. 2012.
- N. Ziviani. Projeto de Algoritmos com Implementação em Java e C++. Cengage Learning. 2006.
- N. Ziviani. Projeto de Algoritmos com Implementação em Pascal e C. 3ed. Cengage Learning. 2010.
- A. Drozdek. Estruturas de Dados e Algoritmos em C++. Cengage. 2009.
- D. E. Knuth. Art of Computer Programming: Sorting and Searching. Addison-Wesley. 1998.
- R. Lafore. Data Structures and Algorithms in Java. 2ed. Sams. 2002.
- R. Sedgewick. Algorithms in Java. 3ed. Addison-Wesley. 2002.
- A. Drozdek. Data Structures and Algorithms in Java. Cengage. 3ed. 2008.

Aulas

- ATLab 103 (Lab SO)
- Atendimento aos alunos: sala CCGT 1102
sex 13h30 às 14h30 (enviar um e-mail)
- email: tiemi@ufscar.br
- material: google Classroom: qbfve6
Programação e Estruturas de Dados (PED)

Avaliações

- T – média de trabalhos práticos
- S – Seminário – apresentação e resumo (final do semestre)
- A – média das avaliações $((A_1 + A_2)/2)$
 A_1 (05/05), A_2 (23/06)
 $MF = 0.2 * T + 0.1 * S + 0.7 * A$, se $A \geq 5,0$ ou
 $MF = A$, caso contrário

 $A = MF > 8.5$
 $B = 7.5 \leq MF \leq 8.5$
 $C = 6.0 \leq MF < 7.5$