



IAA - Aula zero

Fábio Nakano



Fábio Nakano

fabionakano@usp.br (para assuntos relacionados à disciplina, pelo chat do e-disciplinas é melhor)

A1 - 204E

Lab 8

Sou exigente e posso estar de mau humor

Tendo a apresentar os assuntos muito rápido. Neste aspecto, acho melhor se vocês fizerem perguntas, o que torna a aula mais interessante.




Site, ementa e bibliografia

site da disciplina: <https://edisciplinas.usp.br/course/view.php?id=123067#section-0>

ementa: <https://portalservicos.usp.br/iupiterweb/obterDisciplina?nomdis=&sgldis=ach2002>

CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. “Algoritmos - Teoria e Prática”, GEN LTC, 3ª edição, 2012.

ZIVIANI, N. “Projeto de Algoritmos”, Cengage Learning, 3ª edição revista e ampliada, 2011.



Visão geral do assunto da disciplina



ACH2002

Conhecimento prévio

Técnicas de construção de algoritmos

Demonstração de corretude de algoritmos

Análise de algoritmos

Algoritmos e problemas que serão estudados



Conhecimento prévio

Alguma experiência em alguma linguagem de programação imperativa e procedural

Sequências e séries matemáticas (Progressão aritmética, progressão geométrica, somatórios,...)

Análise combinatória básica (Fatorial, permutações, ...)

Cálculo de probabilidades básico (contagem, frequência, probabilidade,...)

Logaritmos (e exponenciais)

Crescimento de funções (inequações)

Derivadas e Limites

Aritmética de matrizes



Técnicas de construção de algoritmos

Definição de problema (e solução) computacional

Algoritmos gulosos (greedy)

Algoritmos por divisão e conquista (divide and conquer)

Programação Dinâmica (dynamic programming)

Tentativa e erro (backtracking)



Demonstração da corretude de algoritmos

Cada algoritmo tem sua demonstração de corretude (não há (ou ainda não descobriram) técnica geral);

Geralmente a demonstração baseia-se em *invariantes* e é feita por *indução matemática*

Nesta disciplina as demonstrações de corretude serão menos formais



Análise de algoritmos

Paradigma de computação: *Random Access Machine*

Principais recursos da máquina: tempo de execução e quantidade de memória

Técnica a aprender: análise assintótica (destaque para a aplicação do Teorema Mestre em algoritmos de divisão e conquista)



Algoritmos e problemas que serão estudados

Insertion, counting, bubble sort

Problema do troco

Merge sort

Problema da mochila (binária)

Heap sort

Problema do caixeiro viajante

Quick sort

Problema do corte de barra

Counting sort

Problema da ordem de
multiplicação de matrizes

Radix sort