SCC 124 - Introdução à Programação para Engenharias

Apresentação

Professor: André C. P. L. F. de Carvalho, ICMC-USP
Posdoutorando: Isvani Frias-Blanco
Monitor: Henrique Bonini de Britto Menezes

Aula de hoje

- Introdução
- Objetivos
- Curso
- Computação e Engenharia de Produção
- Avaliação
- Bibliografia e Ferramentas

Computação

- Combinação de:
  - Matemática
    - Usa linguagens formais para representar ideias
  - Ciências Naturais
    - Observa o comportamento de sistemas complexos, forma hipóteses e testa previsões
  - Engenharia
    - Projeta soluções, combinando componentes e sistemas e verificando vantagens entre alternativas

Objetivo no Jupiter

- Familiarização com os conceitos básicos de:
  - Computadores e da computação
  - Resolução algorítmica de problemas propostos
  - Linguagens de programação de alto nível com aplicações numéricas e não numéricas
- Oferecendo um primeiro contato
  - Com o uso de computadores para desenvolvimento de programas
  - Com os problemas da computação em geral

Como...

- Apresentar, com enfoque prático:
  - Ferramentas e os conceitos básicos de programação de computadores
- Ensinar projeto e desenvolvimento de programas
  - Utilizando técnicas básicas de programação estruturada
- Familiarizar o aluno com o uso das ferramentas necessárias para isso

Conteúdo

- Princípios de computação:
  - Computador
  - Problemas, algoritmos e programas
- Programação de computadores:
  - Linguagens
  - Python
  - Comandos
  - Funções
  - Estruturas de dados

O que é um programa?

- Juntar peças de lego
  - Entrada
  - Saída
  - Matemática
  - Execução condicional
  - Repetição

Áreas da Computação

- Sociedade Brasileira de Computação
- Grandes grupos
  - Aspectos teóricos e fundamentais
  - Computação direcionada a dados
  - Sistemas computacionais
  - Sistemas de software

Áreas da Computação

- Aspectos teóricos e fundamentais
  - Análise de algoritmos
    - Verifica complexidade de algoritmos
  - Análise numérica
    - Resolução de problemas matemáticos usando a computação
  - Teoria da computação
    - O que é computável
  - Lógica

Áreas da Computação

- Computação direcionada a dados
  - Biologia computacional
  - Inteligência artificial
  - Ciência de dados
  - Processamento digital de imagens

Áreas da Computação

- Sistemas computacionais
  - Arquitetura e organização de computadores
  - Redes de computadores
  - Sistemas operacionais
  - Segurança

Áreas da Computação

- Sistemas de software
  - Banco de dados
  - Engenharia de software
  - Interação humano-computador
  - Linguagens de programação

Computação na Produção

- Várias áreas da Engenharia de Produção
  - Recomendação
    - Recursos Humanos
  - Predição
    - Chão de fábrica
  - Otimização Combinatória / Pesquisa Operacional
    - Logística

Predição

- Predizer o valor de saída para uma dada entrada
  - Diagnóstico de falhas em peças
  - Finanças
    - Análise de crédito
    - Predição de falências
    - Predição de compra/venda de ações

Pesquisa Operacional

- Busca pela melhor solução para um problema com várias soluções
  - Reduzir custo ou aumentar satisfação
  - Algoritmos exaustivos
    - Tenta todas as soluções
  - Algoritmos exatos
    - Busca solução por meio de prova matemática
  - Heurísticas
  - Metaheurísticas

Heurísticas

- Humanos utilizariam intuição ou palpite para acelerar busca por melhores soluções
  - Imagine que você está no centro da sua cidade e
    - Quer pegar um trem para casa, que fica em um bairro B, mas não sabe qual deve pegar
    - Solução simples
      - Se mora na zona Norte, ignorar trens que vão para o sul
      - Se mora na zona Sul, ignorar trens que vão para o Norte
      - Heurísticas ajudam a limitar a busca
  - Intuições ou palpites são chamados de heurísticas

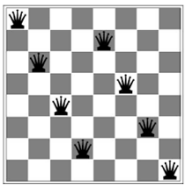
Meta-heurísticas

- Uma heurística é uma forma de resolver um problema de específico
- Meta-heurística é uma forma de resolver vários problemas usando intuição ou dica
  - Inspiradas na natureza
    - Algoritmos Genéticos
    - Otimização Baseada em Colônias de Formigas
    - Sistemas Imunes Artificiais
    - Computação baseada em DNA

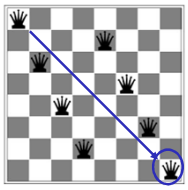
Problemas clássicos

- Encontrar um caminho para um alvo
- Missionários e canibais
- Travessia de rio
- Jogos
  - Xadrez
  - Jogo de gamão
  - Torres de Hanói
- N-rainhas

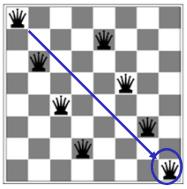
Exemplo: N-Rainhas

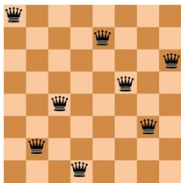


Exemplo: N-Rainhas



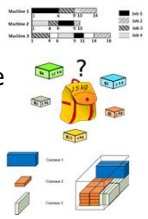
Exemplo: N-Rainhas





Pesquisa Operacional

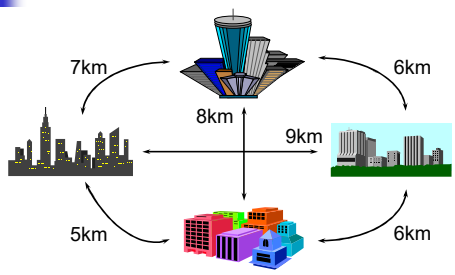
- Alocação ótima de recursos
  - Alocar tarefas e máquinas a operadores
    - *Job Shop Schedule (JSS)*
- Definição do melhor trajeto
  - Problema do caixeiro viajante
    - *Traveling Salesman Problem (TSP)*
- Problema da mochila
  - *Knapsack (ruchsack) Problem*
- Problema de corte



Problema caixeiro viajante (TSP)

- Caixeiro viajante pretende visitar N cidades
  - Cada cidade uma única vez
  - Percorrer menor distância
    - Cada visita tem um custo proporcional à distância percorrida
  - Voltar à cidade inicial
- Aparentemente simples

TSP



TSP

- Simplicidade da definição do problema engana
  - NP-completo
  - Um dos problemas de otimização mais estudados
  - Não existe nenhuma solução eficiente para o caso geral
    - Quem resolver este problema para o caso geral receberá um prêmio de US 1000000 do *Clay Mathematics Institute*

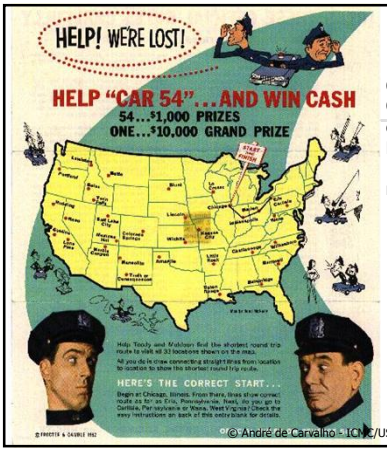
Vídeo

TSP

#Cidades	# soluções (n-1)!	Tempo total estimado em PC
5	24	Insignificante
10	362.880	0,003 s
15	87 bilhões	20 min
20	$1{,}2 \times 10^{17}$	73 anos
25	$6{,}2 \times 10^{23}$	470 milhões de anos

TSP

- Histórico
  - Problemas matemáticos relacionados ao caixeiro viajante foram estudados por volta de 1800
    - Pelos matemáticos Sir William Rowan Hamilton e Thomas Penyngton Kirkman
  - Problema geral foi primeiro tratado por volta de 1930 por Karl Menger

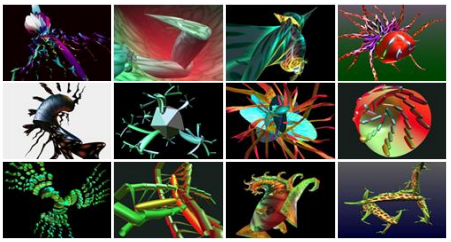


Concurso da Proctor and Gamble, em 1962, para encontrar melhor rota Percorrendo 33 cidades (distâncias tiradas do mapa Rand McNally)

Desafio TSP: Monalisa

- Exemplo de TSP com 100.000 pontos
  - Desenhar linha contínua
  - Cidades visitadas em sequência são ligadas por uma linha

Desenho por computador



http://www.genarts.com/galapagos/galapagos-images.html

Música por computador

- Algoritmo genético que aprende a compor música
  - Mesmo estilo usado por Bach
  - Jazz

Etiqueta em sala de aula

- Chegar no horário da aula
- Pedir licença para entrar e sair da sala
- Usar palavras "mágicas" por favor, com licença, obrigado (a) e desculpe
- Não conversar durante a aula
- Levantar o braço para fazer perguntas e comentários
- Não ler outro material durante a aula
- Desligar celular durante a aula
- Colocar lixo no lixo
- Não copiar de colega ou internet material a ser avaliado

Código de Honra

- Espera-se que cada aluno obtenha seu grau baseado exclusivamente na avaliação de seu esforço e trabalho pessoal
  - Qualquer forma de conversa em exames, ou plagiarismo em trabalhos, constitui-se em fraude
- Comportamentos não são aceitos na USP
  - Atitudes previstas no Código Disciplinar Discente desta Universidade e são puníveis de acordo com o grau de severidade de acordo com esta regulamentação
- O código de honra deverá ser rigorosamente seguido sob pena de reprovação

Andamento

- Aprender a aprender
  - Promover a cultura do conhecimento, ao invés de meramente transmitir informação
- Professor é um facilitador
  - Aluno precisa ler material do curso e praticar programação
  - Desempenho do aluno depende de sua dedicação

Aulas laboratório

- Aluno deve ter estudado os conceitos vistos na aula teórica
  - Não se espera que o aluno pergunte sobre conceitos que estão nos slides
- Só se aprende a programar programando

Aulas laboratório

- Resposta a perguntas
  - " Como eu faço?"
    - Espera-se que o aluno crie uma solução utilizando os conceitos aprendidos nas aulas
  - " Meu programa esta certo?"
    - Aluno deve testar suas ideias e verifica o que provoca no programa
    - Só quem pode responder bem é o interpretador Python ou rastreamento do código pelo aluno

Aulas laboratório

- Verificar se o programa esta certo
  - Erro de sintático: estudar material das aulas teóricas
  - Erro de execução
    - Rastrear código
    - Na mão ou imprimindo resultados parciais

Avaliação

- Provas:
  - Haverá duas provas
  - 5/4 e 20/6
  - Cada prova vale de 0 a 10
- Trabalhos:
  - Haverá um ou mais trabalhos práticos
    - Cada trabalho vale de 0 a 10

Avaliação

- Cálculo da média:
  - MP = Média Aritmética das Provas
  - MT = Média Aritmética dos Trabalhos
  - MF = Média Final
    - Se MP ≥ 5 e MT ≥ 5 → MF = (7MP + 3MT) /10
    - Se MP < 5 ou MT < 5 → MF = menor valor entre MP e MT
- Fraude: média zero

Avaliação

- Recuperação:
  - Só terão direito à recuperação os alunos com **3.0 ≤ MF ≤ 5.0** e frequência superior a 70%

Sobre o Curso

- Livros (gratuitos e com tradução)
  - Deve ser Python 3
  - Byte of Python

    - Swaroop, C. H.
    - *http://www.swaroopch.com/notes/python/*
  - Think Python, 2nd edition

    - How to Think Like a Computer Scientist
    - Downey, A. B.
    - *http://www.greenteapress.com/thinkpython/*

Ferramentas

- Curso prático
- Python 3.6.0
  - Anaconda
  - Gratuito

- PyCharm
  - Gratuita

Conclusão

- Como vai ser o curso
- Objetivos e conteúdo
- Computação e Engenharia de Produção
- Avaliação
- Material bibliográfico
- Ferramenta

Perguntas