



www.python.pro.br

Roteiro - Paradigmas de Progr.

Ganancioso (Greedy)

Divisão e Conquista

Programação Dinâmica

Last Resource Used

Exercício*

Ganancioso, Guloso (Greedy)

Chega-se em um mínimo local

Aplica-se iterativamente esse mínimo

Chega-se em mínimo global

Deve ser demonstrado que mínimos locais
levam a mínimo global

Ex: algoritmo de Dijkstra *

Divisão e Conquista

Problema é dividido em instância de problemas menores - Divisão

Caso trivial é resolvido - Conquista

Subsoluções são fundidas para chegar em resposta

Exemplo: Merge e Quick Sort*

Programação Dinâmica

Relacionado com Divisão e Conquista

Utilizado para minimizar tempo de execução às custas de memória

Solução já encontradas são armazenadas em uma estrutura, como uma lista ou mapa

Essa estrutura costuma ser chamada de cache

Ex: fibonacci inocente *

Iru

Lib do Python que cria o mapa para vc

Argumentos precisam ser *hashable* por que serão utilizados como chave de um dicionário

É chamado de decorator

Explicar decorator se houver tempo e curiosidade *

Exercício - Mínima Soma Quad

Encontrar menor soma de quadrados perfeitos cuja soma seja n

Ex: 1: [1], 2: [1,1], 3: [1,1,1], 11:[9,1,1], 12:[4,4,4]

Solução gananciosa não funciona, ver caso 12

Utilizar programação dinâmica

Não precisa calcular complexidade

Testes em <http://bit.ly/26AGZq6> *

Obrigado

renzo@python.pro.br
@renzoprobr

