

CONJUNTOS

MC102 - Algoritmos e
Programação de
Computadores

Santiago Valdés Ravelo
[https://ic.unicamp.br/~santiago/
ravelo@unicamp.br](https://ic.unicamp.br/~santiago/ravelo@unicamp.br)

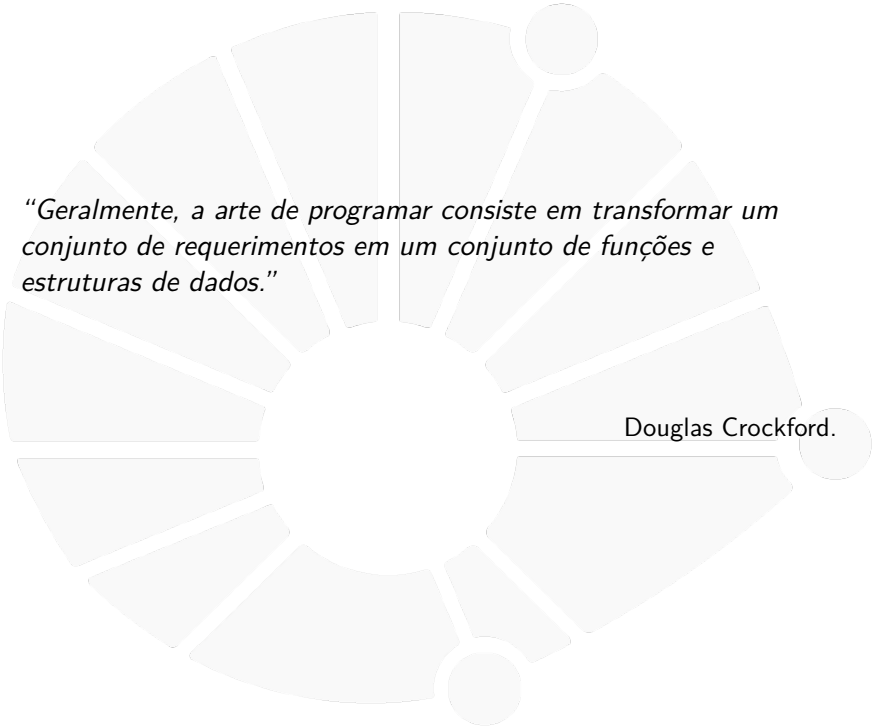
04/24

14



UNICAMP





“Geralmente, a arte de programar consiste em transformar um conjunto de requerimentos em um conjunto de funções e estruturas de dados.”

Douglas Crockford.



DÚVIDAS DA AULA ANTERIOR



Dúvidas selecionadas

- ▶ Você poderia dar uma explicaçãozinha breve sobre o problema $P = NP$, fiquei curioso.
- ▶ Digamos que tenho um `dict1=dict(carro1="Fiat", carro2="Ford")` e um `dict2=dict(carro2="Ford", carro1="Fiat")`. Se não há uma "ordem" específica dos pares chave-valor nos dicionários, os dicionários 1 e 2 são idênticos, correto? Então um `print(dict1==dict2)` deveria retornar True?
- ▶ Eu entendi como os dicionários são organizados no código, mas se for um dicionário com muitas entradas, não vai ficar uma linha muito grande e meio difícil de ler? por ex: `d = {'a':'b', 'c':'d', 'e':'f', ..., 'y':'z'}`.
- ▶ Sabendo que, se eu imprimir um dicionário, ele vem na ordem de inserção, eu posso acessar essa ordem de inserção de alguma forma para usar em algoritmos?
- ▶ Todos aqueles comandos que você passou sobre lista também se aplicam ao dicionário? como o `clear`, `slice` e `tals`?
- ▶ Dicionários são mutáveis? Se sim, então para armazenar em outra variável será preciso fazer tipo `dict.copy()` como nas listas?
- ▶ Nos dicionários, é possível usar o `.pop()` igual nas listas? E se for possível, ele retorna o valor que estava na chave?
- ▶ Tem algum limite claro de até onde podemos fazer uma lista de lista de lista... ou o mesmo com dicionários?
- ▶ Em quais casos seria melhor usar uma lista ou um dicionário? Como vou conseguir saber diferenciar o momento de usar um ou outro?
- ▶ Foi dito que caso as chaves de um dicionário são todas strings pode não se utilizar de aspas para defini-los, porém caso a chave tenha uma letra junto de números ele ainda é uma string?
- ▶ Não entendi muito bem como funciona o método de `keys()` no dicionário, poderia explicar de novo por favor?
- ▶ É possível adicionar mais que pares em um dicionário? Por exemplo, um trio de informações ou algo do tipo.
- ▶ Como nas listas, é possível fazer algo como `dict = { 'chave': 'valor' } * 5`?



CONJUNTOS



O que são?

Conjuntos lembram dicionários:

- ▶ Porém, ao invés de armazenar pares chave-valor.
- ▶ Armazenam apenas chaves.
- ▶ As chaves podem ser **int**, **str**, **float**.
- ▶ E até outros objetos... mas não todos...

Como na matemática, conjuntos não têm repetição!



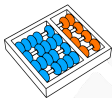
Criando conjuntos

Conjuntos com valores iniciais:

- ▶ `s = {"ana", "beto", "carlos"}`.
 - ▶ Notação similar a do dicionário.
 - ▶ Mas sem os valores, apenas chaves.
- ▶ `s = set(["ana", "beto", "carlos"])`.

Conjuntos vazios:

- ▶ `s = set()`.
- ▶ Não podemos escrever `s = {}`.
 - ▶ Porque isso cria um dicionário vazio!



Manipulando conjuntos

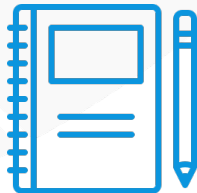
- ▶ **s.add(x)** adiciona **x**:
 - ▶ Apenas se **x** não estiver no conjunto.
- ▶ **del** não funciona... (conjuntos não tem pares chave-valor).
- ▶ **s.discard(x)** remove **x**:
 - ▶ Se **x** não está em **s**, nada acontece.
- ▶ **s.remove(x)** remove **x**:
 - ▶ Se **x** não está em **s**, temos um **KeyError**.
- ▶ **x in s** para verificar se **x** está em **s**.
- ▶ **for x in s:** para iterar sobre **s**:
 - ▶ A ordem de acesso não é necessariamente a ordem de inserção no conjunto!



Conjuntos



Vamos fazer alguns exercícios?





Exercícios

1. Faça uma função que, dados conjuntos $s1$ e $s2$, devolve um novo conjunto representando a união de $s1$ e $s2$.
2. Faça uma função que, dados conjuntos $s1$ e $s2$, devolve um novo conjunto representando a interseção de $s1$ e $s2$.
3. Faça uma função que, dados conjuntos $s1$ e $s2$, devolve um novo conjunto representando a subtração de $s2$ em $s1$.
4. Faça uma função que, dados conjuntos $s1$ e $s2$, devolve se $s1$ é subconjunto de $s2$.
5. Faça uma função que, dada uma lista, devolve uma nova lista sem elementos repetidos. Dica: use conjuntos!



Métodos úteis de conjunto

Devolvem um novo conjunto:

- ▶ **`s1.union(s2)`**: união.
 - ▶ Ou $s1 \mid s2$.
 - ▶ O símbolo \mid representa **ou**.
- ▶ **`s1.intersection(s2)`**: interseção.
 - ▶ Ou $s1 \& s2$.
 - ▶ O símbolo $\&$ representa **e**.
- ▶ **`s1.difference(s2, ...)`**: diferença.
 - ▶ Ou $s1 - s2$.
- ▶ **`s1.symmetric_difference(s2)`**: diferença simétrica.
 - ▶ Ou $s1 \hat{\ } s2$.
 - ▶ O símbolo $\hat{\ }$ representa **ou exclusivo**.



Métodos úteis de conjunto

Alteram o próprio conjunto:

- ▶ **`s1.update(s2)`**: altera **`s1`** para a união de **`s1`** e **`s2`**.
 - ▶ Ou **`s1 |= s2`**.
- ▶ **`s1.intersection_update(s2)`**: interseção.
 - ▶ Ou **`s1 &= s2`**.
- ▶ **`s1.difference_update(s2, ...)`**: diferença.
 - ▶ Ou **`s1 -= s2`**.
- ▶ **`s1.symmetric_difference_update(s2)`**: diferença simétrica.
 - ▶ Ou **`s1 ^= s2`**.



Métodos úteis de conjunto

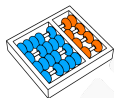
Relações:

- ▶ **`s1.isdisjoint(s2)`**: são disjuntos?
- ▶ **`s1.issubset(s2)`**: **`s1`** é subconjunto de **`s2`**?
 - ▶ Ou **`s1 <= s2`**.
- ▶ **`s1.issuperset(s2)`**: **`s1`** é superconjunto de **`s2`**?
 - ▶ Ou **`s1 >= s2`**.

Como sempre, veja a documentação para mais métodos e detalhes!



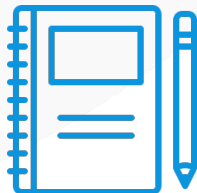
EXERCÍCIOS

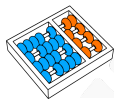


Conjuntos



Vamos fazer alguns exercícios?





Voluntários?





Exercícios

1. Faça um programa que encontre o mínimo e o máximo em um dado conjunto de números.
2. Faça uma função que dados dois conjuntos de números inteiros, retorna dois novos conjuntos: um com os dobros dos elementos repetidos e outro com a metade (inteira) dos elementos não repetidos.
3. Faça uma função que dados dois conjuntos, retorna um conjunto contendo qualquer valor que pode ser obtido como soma de um elemento do primeiro com um elemento do segundo.
4. Faça uma função que recebe um conjunto de números e retorna o maior produto entre dois números do conjunto.
5. Faça uma função que recebe um conjunto de inteiros **S** e um inteiro **x** e retorna um conjunto contendo todos os pares de valores de **S** que somados resultam em **x**.
6. Faça uma função que recebe um conjunto de strings e retorna o maior prefixo de todas as strings do conjunto.

CONJUNTOS

MC102 - Algoritmos e
Programação de
Computadores

Santiago Valdés Ravelo
[https://ic.unicamp.br/~santiago/
ravelo@unicamp.br](https://ic.unicamp.br/~santiago/ravelo@unicamp.br)

04/24

14



UNICAMP

