情報システム 2 学期

2024 年 9 月 11 日

1. データ構造とアルゴリズム

1.1. データの、型と列データ

1.1.1. 基本的なデータ型

1.1.1.1. 論理型

否定		論理積			論理和		
X	NOT X	X	Y	X AND Y	X	Y	X OR Y
偽	真	偽	偽	偽	偽	偽	偽
真	偽	偽	真	偽	真	偽	真
		真	偽	偽	真	偽	真
		真	真	真	真	真	真

1.1.1.2. 文字列型

Unicode
…文字をコンピュータで扱うための標準規格。
　世界で使われる多くの文字を扱えるように策定されている Unicode が、文字集合（文字コード）として広く採用されている。

1.1.1.3. 数値型

・実数型は、 **浮動小数点数** で表現される。
・2 進法で限られたビット数によって表現するため、表現できる数値の範囲が限られる。

1.1.2. 文字列

文字列
...文字を一列に並べたもの。
　文字列は、文字の配列として扱われることが多い。

C	o	m	p	u	t	e	r
0	1	2	3	4	5	6	7

文字列の長さ
...文字列の長さは、文字列に含まれる文字の数で表現される。
　例えば、"Computer" は 8 文字の長さを持つ。
　このとき、先頭の文字の添字は 0 であり、これを 0-origin という。

```
>>> s = 'Computer'
>>> s
'Computer'

>>> s[2]
'm'

>>> type('AAA')
<class 'str'>
```

1.1.2.1. 2 つの文字列の連結

```
>>> 'ABC' + 'DEF'
'ABCDEF'
```

1.1.2.2. 2 つの文字列の比較

```
>>> 'ABC' < 'ABE' # 小さい方が先
True

>>> 'AB' < 'ABA'  # 長い方が大きい
True
```

1.1.2.3. 部分文字列の取得

```
>>> 'ABCDEF'[2:4]
'CD'

>>> s = 'ABCDEF'
>>> s[2:4]
'CD'
```

1.1.3. 列データ構造

データ構造とはデータの集まりをコンピュータの中で効果的に扱うための形式のこと。

1.1.3.1. リスト

・リストは、複数のデータをまとめて扱うためのデータ構造。
・リストの要素は、インデックスを使ってアクセスできる。

1.1.3.2. 配列

・配列は、同じ型のデータを連続したメモリ領域に格納するデータ構造。
・配列の要素は、インデックスを使ってアクセスできる。

1.1.3.3. スタック

・スタックは、データを一時的に保存するためのデータ構造。
・データの追加や削除は、常に一方向から行われる。

```
>>> data = [4, 1, 7, 3, 2]
>>> data.append(8)

>>> data
[4, 1, 7, 3, 2, 8]

>>> data[-1]
8

>>> data.pop()
8

>>> data
[4, 1, 7, 3, 2]
```

1.1.3.4. キュー

・キューは、データを一時的に保存するためのデータ構造。
・データの追加は一方向から行い、削除は反対方向から行う。

```
>>> from collections import deque
>>> data = deque([4, 1, 7, 3, 2])

>>> data
deque([4, 1, 7, 3, 2])

>>> data.popleft()
4

>>> data
deque([1, 7, 3, 2])
```

```
>>> data.append(5)
>>> data
deque([1, 7, 3, 2, 5])

>>> data.appendleft(6)
>>> data
deque([6, 1, 7, 3, 2, 5])
```