

Řešení domácích úkolů - 5. týden

Nejdelší souvislá rostoucí podposloupnost

Ze standardního vstupu načtete obvyklým způsobem (jedno číslo na řádek, ukončená -1) neprázdnou posloupnost celých čísel a najděte nejdelší striktně rostoucí podposloupnost následujících čísel. Vytiskněte tuto podposloupnost na standardní výstup jako seznam, tedy i s hranatými závorkami.

Příklad 1

Vstup:

1
2
3
4
5
6
-1

Výstup:

[1,2,3,4,5,6]

Příklad 2

Vstup:

2
-2
-1

Výstup:

[2]

Příklad 3

Vstup:

1
1
2
1
2
3

2
3
4
5
4
5
6
5
6
5
-1

Výstup:

[2,3,4,5]

Poznámky

- ReCodEx-ové testy pro tuto úlohu:
 - Posloupnosti z příkladů 1-3
 - Náhodná posloupnost 1000 čísel vygenerovaná z čísel 1 - 10 markovovským procesem, který vytváří mnoho rostoucích sekvencí. Přejchodová matice procesu je uvedena v následující tabulce, $q = 0.3$, $p = 1 - q$.

| | | | | | | | | | | | |
|---|----|---|---|---|---|---|---|---|---|---|----|
| ◦ | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| | 1- | q | p | | | | | | | | |
| | 2 | q | | p | | | | | | | |
| | 3 | | q | | p | | | | | | |
| | 4 | | | q | | p | | | | | |
| | 5 | | | | q | | p | | | | |
| | 6 | | | | | q | | p | | | |
| | 7 | | | | | | q | | p | | |
| | 8 | | | | | | | q | | p | |
| | 9 | | | | | | | | q | | p |
| | 10 | p | | | | | | | | q | |

(Prvek v poli (i,j) je pravděpodobnost přechodu z i do j v libovolném kroku.)

- Dejte pozor při kopírování seznamů.
- Pro libovolnou neprázdnou vstupní posloupnost je řešením neprázdný seznam.

- Upřednostněna budou řešení, která neukládají celou vstupní posloupnost do paměti.

Analýza

U této úlohy se budeme zabývat pouze sekvenčním řešením, tedy řešením, které neukládá celou posloupnost od paměti. Důvod je ten, že při načtení posloupnosti do paměti nezískáváte žádnou výhodu a budete nejspíš postupovat stejně jako u sekvenčního řešení.

Pro sekvenční řešení potřebujeme definovat stav, který budeme postupně aktualizovat při načítání jednotlivých hodnot posloupnosti. Tento stav budou tvořit dvě posloupnosti:

- aktuální rostoucí posloupnost - nově načtená hodnota se do ní připojí, pokud je větší než poslední hodnota posloupnosti. V opačném případě se posloupnost vyprázdní a uloží se do ní nově načtená hodnota.
- nejdelší dosud nalezená posloupnost - aktualizuje se, je-li načtená hodnota menší než poslední hodnota aktuální posloupnosti. V takovém případě se aktuální posloupnost uloží do nejdelší posloupnosti, pokud je delší. Nejdelší posloupnost se také aktualizuje po ukončení načítání.

Obě posloupnosti jsou inicializovány jako prázdné seznamy.

Vzorové řešení

```
1  MIN_VALUE = -1.0e-10
2  current_seq = []
3  max_seq = []
4  prev = MIN_VALUE
5
6  while (current := int(input())) != -1:
7      if current > prev:
8          current_seq.append(current)
9      else:
10         if len(current_seq) > len(max_seq):
11             max_seq = current_seq.copy()
12             current_seq = [current]
13         prev = current
14
15 if len(current_seq) > len(max_seq):
16     print(current_seq)
17 else:
18     print(max_seq)
```

Obvyklé problémy

Tato úloha je obvykle vnímána jako lehčí. Poměrně často se ale vyskytují řešení, které z nějakého důvodu předpokládají, že sousedící hodnoty v rostoucí posloupnosti se budou lišit o 1.

Pravděpodobný důvod je poslední příklad s posloupností, generovanou Markovovským procesem. Tento předpoklad není oprávněný.