



FAKULTA
INFORMATIKY
Masarykova univerzita

IB111 Základy programovania

Cvičenie 7 – radiace algoritmy

Matej Troják

(Poďakovanie: Valdemar Švábenský)

1. Bubble sort

2. Selection sort

3. Insertion sort

Bubble sort

- Vstup: zoznam hodnôt, ktoré sa dajú usporiadať
- Algoritmus prechádza zoznam od začiatku a porovná **každú dvojicu susedných prvkov**
 - Ak sú susedia v opačnom poradí, vymení ich
 - Ak sú susedia v správnom poradí, nechá ich tak
- Ak nastala aspoň jedna výmena, zoznam sa bude prechádzať **znova od začiatku**
- Výstup: usporiadaný zoznam hodnôt vstupného zoznamu

Bubble sort

0	1	2	3	4	5	6	7	8
23	17	5	90	12	44	38	84	77

 exchange

17	23	5	90	12	44	38	84	77
----	----	---	----	----	----	----	----	----

 exchange

17	5	23	90	12	44	38	84	77
----	---	----	----	----	----	----	----	----

 exchange
ok

17	5	23	12	90	44	38	84	77
----	---	----	----	----	----	----	----	----

 exchange

17	5	23	12	44	90	38	84	77
----	---	----	----	----	----	----	----	----

exchange 

17	5	23	12	44	38	90	84	77
----	---	----	----	----	----	----	----	----

exchange 

17	5	23	12	44	38	84	90	77
----	---	----	----	----	----	----	----	----

exchange 

17	5	23	12	44	38	84	77	90
----	---	----	----	----	----	----	----	----

The largest value 90 is at the end of the list.

Úloha 1 – Bubble sort

- Napíšte funkciu `bubble_sort(lst)`, ktorá pomocou algoritmu bubble sort zoradí zoznamý `lst` daný ako parameter. Využite testovacie výpisy pred každou iteráciou cyklu pre kontrolu priebehu radenia.

```
>>> bubble_sort([13, 8, 11, 3, 5])
[13, 8, 11, 3, 5]
[8, 11, 3, 5, 13]
[8, 3, 5, 11, 13]
[3, 5, 8, 11, 13]
Finish: [3, 5, 8, 11, 13]
>>> bubble_sort([9, 8, 4, 6, 1])
[9, 8, 4, 6, 1]
[8, 4, 6, 1, 9]
[4, 6, 1, 8, 9]
[4, 1, 6, 8, 9]
[1, 4, 6, 8, 9]
Finish: [1, 4, 6, 8, 9]
```

1. Bubble sort

2. Selection sort

3. Insertion sort

Selection sort

- Vstup: zoznam hodnôt, ktoré sa dajú usporiadať
- Vstupný zoznam je **pomyselne** rozdelený na dve časti: *zoradenú* a *nezoradenú*
- Najprv je *zoradená* časť prázdna a *nezoradená* časť je celý vstupný zoznam
- V každom kroku algoritmus nájde **minimum** *nezoradenej* časti a vymení ho s najľavejším prvkom *nezoradenej* časti
 - Tým ho vlastne pridá na koniec (napravo) *zoradenej* časti
 - *Zoradená* časť sa buduje zľava doprava
- Keď je *nezoradená* časť prázdna, algoritmus končí
- Výstup: usporiadaný zoznam hodnôt vstupného zoznamu

Selection sort



Úloha 2 – Selection sort

- Napíšte funkciu `selection_sort(lst)`, ktorá pomocou algoritmu selection sort zoradí zoznamý `lst` daný ako parameter. Využite testovacie výpisy pred každou iteráciou cyklu pre kontrolu priebehu radenia.

```
>>> selection_sort([13, 8, 11, 3, 5])
```

```
[13, 8, 11, 3, 5]
```

```
[3, 8, 11, 13, 5]
```

```
[3, 5, 11, 13, 8]
```

```
[3, 5, 8, 13, 11]
```

```
[3, 5, 8, 11, 13]
```

```
Finish: [3, 5, 8, 11, 13]
```

```
>>> selection_sort([9, 8, 4, 6, 1])
```

```
[9, 8, 4, 6, 1]
```

```
[1, 8, 4, 6, 9]
```

```
[1, 4, 8, 6, 9]
```

```
[1, 4, 6, 8, 9]
```

```
[1, 4, 6, 8, 9]
```

```
Finish: [1, 4, 6, 8, 9]
```

1. Bubble sort

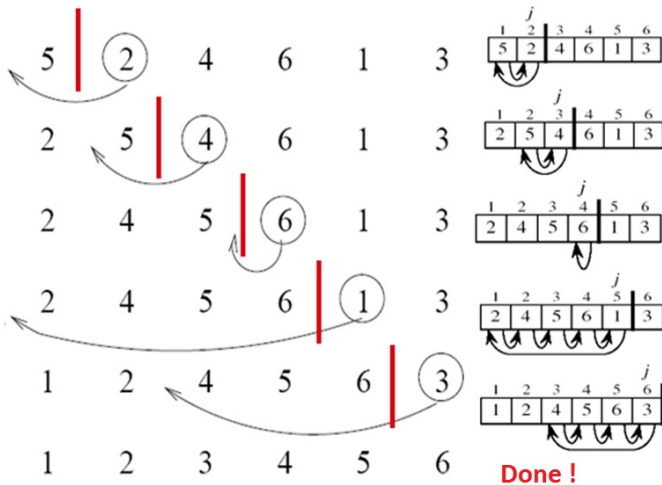
2. Selection sort

3. Insertion sort

Insertion sort

- Vstup: zoznam hodnôt, ktoré sa dajú usporiadať
- Vstupný zoznam je **pomyselne** rozdelený na dve časti: *zoradenú* a *nezoradenú*
- Najprv *zoradená* časť obsahuje prvý prvok vstupného zoznamu a *nezoradená* časť je zvyšok zoznamu
- V každom kroku algoritmus vezme prvý prvok *nezoradenej* časti a vloží ho na správne miesto *zoradenej* časti
- Keď je *nezoradená* časť prázdna, algoritmus končí
- Výstup: usporiadaný zoznam hodnôt vstupného zoznamu

Insertion sort



Úloha 3 – Insertion sort

- Napíšte funkciu `insert_sort(lst)`, ktorá pomocou algoritmu insertion sort zoradí zoznamý `lst` daný ako parameter. Využite testovacie výpisy pred každou iteráciou cyklu pre kontrolu priebehu radenia.

```
>>> insert_sort([13, 8, 11, 3, 5])
[13, 8, 11, 3, 5]
[8, 13, 11, 3, 5]
[8, 11, 13, 3, 5]
[3, 8, 11, 13, 5]
Finish: [3, 5, 8, 11, 13]
>>> insert_sort([9, 8, 4, 6, 1])
[9, 8, 4, 6, 1]
[8, 9, 4, 6, 1]
[4, 8, 9, 6, 1]
[4, 6, 8, 9, 1]
Finish: [1, 4, 6, 8, 9]
```