# 19. Řadící algoritmy I (Bubble, Select, Insert)

Řadící algoritmy zajišťující seřazení daného souboru dat do specifikovaného pořadí. Nejčastěji se řadí podle velikosti čísel nebo abecedně.

## Bubble Sort

V poli se porovnávají 2 sousední prvky.

Pokud je číslo vlevo větší (menší) než vpravo, prohodí se.

Po průchodu celým polem je seřazený minimálně 1 prvek. Pole se o seřazený prvek zkrátí a pokračuje znova, dokud pole není seřazeno.

Počet porovnání = (n2 - n) / 2

Počet přesunů = přibližně n2

Složitost = n2

#### public static void bubble(int[] array){

#### for (int i = 0; i < array.length-1; i++){

#### for (int j = 0; j < array.length-1-i; j++){

#### //if(array[j] < array[j+1]){ směr řazení

#### if(array[j] > array[j+1]){

#### int tmp = array[j];

#### array[j] = array[j+1];

#### array[j+1] = tmp;

#### }

#### }

#### }

#### }

## Select Sort

Najdeme maximum (minimum) v poli.

Tento prvek se prohodí s prvkem na posledním (prvním) místě.

Postup se opakuje a po každém průchodu bude vynechán poslední (první) prvek.

Počet porovnání = (n2 - n) / 2

Počet přesunů = přibližně n-1

Složitost = n2

#### public static void select(int[] array){

#### for (int i = 0; i < array.length-1; i++){

#### int max = pole.length-i-1;

#### for (int j = 0; j < array.length-i; j++){

#### //if(array[j] < array[max]){ směr řazení

#### if(array[j] > array[max]){

#### max = j;

#### }

#### }

#### int tmp = array[array.length-1-i];

#### array[array.length-1-i] = array[max];

#### array[max] = tmp;

#### }

#### }

## Insert Sort

Postupně prochází prvky a každý nesetříděný se zařadí na správné místo.

K tomu je nutné mít pole již částečně setříděné. To se provede tak, že se první prvek považuje za setříděný a začne se od druhého.

Počet porovnání = (n2 - n) / 2

Počet přesunů = přibližně (n2 – n - 2) / 2

Složitost = přibližně n2

#### public static void insert(int[] array){

#### for (int i = 0; i < array.length; i++){

#### int tmp = array[i];

#### int j = i - 1;

#### //while((j>=0) && (array[j] < tmp)){směř řazení

#### while((j >= 0) && (array[j] > tmp)){

#### array[j+1] = array[j];

#### j--;

#### }

#### array[j+1] = tmp;

#### }

#### }