Упражнения: Сортиране чрез вмъкване и по метода на мехурчето

Problem 1. Сортиране чрез метода на мехурчето

Сортирайте един масив от елементи с помощта на метода на мехурчето (Bubble Sort).

Примери

Вход	Изход
5 4 3 2 1	1 2 3 4 5

Подсказки

Първо разгледайте описанието на алгоритъма от Уикипедия и визуализацията му във Visualgo.net. Или пък изберете по-нестандартната визуализация като унгарски народен танц, представен от AlgoRythmics. ☺

После създайте клас **`BubbleSort`** с един-единствен метод Sort

```
public class BubbleSort
{
    public static void Sort<T>(T[] a) where T : IComparable<T>
    {

    }

}
```

Създайте и два помощни метода - за сравняване на два елемента и за размяна на местата им:

```
static bool Less(IComparable first, IComparable second)
{
    return first.CompareTo(second) < 0;
}

static void Swap<T>(T[] collection, int from, int to)
{
    //TODO: Размяна на елементите
}
```

В реализацията на метода **`Sort()`** **опишете механизъма на** самото сортиране:

- Обхождаме целия масив и **сравняваме всеки 2 съседни елемента**
- Ако първия е по-голям от втория, **те разменят местата си**.
- **Това се повтаря** до подреждането на масива.

Алгоритъмът може да бъде реализиран с вложени цикли:

- Вътрешният цикъл е за сравняване на елементите и размяна на местата им, ако първият е по-голям от втория. В резултат на това обхождане и размяна най-голямото („най-тежкото“) число „потъва“ до „дъното“ (края) на масива, а по-малките („по-леките“) числа „изплуват“ като мехурчета с по една позиция, ако са били след по-големи. Оттук и името на метода. Ако при някое обхождане няма нито една

размяна, значи масивът вече е подреден. Помислете дали не можете да използвате това за оптимизиране на алгоритъма.

- Външният цикъл ни служи да укажем до кой елемент ще сравняваме съседните елементи. Помислете дали всеки път ще е нужно да проверяваме до края на масива.

Problem 2. Сортиране чрез вмъкване

Сортирайте един масив от елементи с помощта на метода на сортиране чрез вмъкване (Insertion Sort). Но първо разгледайте описанието на алгоритъма от Уикипедия и визуализацията му във Visualgo.net или пък тази под формата на румънски народен танц, представен от AlgoRythmics. ☺ Можете да направите сравнение между този и другите алгоритми за сортиране, които познавате, и чрез чудесната визуализация на David Galles от USF .

Примери

| **Вход** | **Изход** |	| **Вход** | **Изход** |
| --- | --- |	| --- | --- |

Подсказки

Използвайте указанията на предната задача. Алгоритъмът накратко:

- Преместваме първия несортиран елемент наляво на мястото му
- Повтаряме това докато не останат несортирани елементи

Problem 3. Визуализация на сортирането

Модифицирайте кода на предните две задачи, така че да служи и за визуализация на сортирането по тези два метода. За целта на конзолата отпечатвайте всяка съществена стъпка от алгоритъма, кои елементи ще се разменят и как изглежда масивът след всяка размяна. Обсъдете в клас коя визуализация се е получила най-прегледна и разбираема и защо.

Подсказки

Модифицирайте метода **Swap( )**, така че да извежда кои елементи се разменят и да отпечатва масива след размяната.

Problem 4. Уравновесен масив

Да се напише програма, която позволява да се въведе размер на масив и стойностите на неговите елементи и после го подрежда така, че стойностите на елементите да нарастват от началото до средата на масива и след това отново да намаляват от средата до края. Елементите от първата половина на масива не трябва да преминават във втората му половина. Полученият масив трябва да бъде отпечатан.

Вход

- Входните данни трябва да се прочетат от конзолата.
- На първия ред се подава цяло четно число N, съдържащо броят на числата в масива
- На втория ред се подават N цели числа, отделени едно от друго с интервал. Това са стойностите на елементите в масива.

- Входните данни винаги ще са валидни и в описания формат. Не е необходимо да бъдат изрично проверявани.

Изход

- Изходните данни („уравновесеният“ масив) трябва да бъдат отпечатани на конзолата.

Пример

Вход	Изход	Коментар
10 **4 2 6 3 8** 1 7 4 2 9	**2 3 4 6 8** 9 7 4 2 1	За яснота елементите от първата половина на масива **са удебелени.**

Problem 5. Сортиране по произволна колона

Напишете програма, която сортира таблица от числови стойности по произволна колона.

Вход

- Входните данни трябва да се прочетат от конзолата.
  - На първия ще ви бъдат подадени три цели числа R, C и S, разделени с интервал. R е броят на редовете в таблицата, C - броят на колоните, а S - номерът на колоната, по която трябва да бъде сортиран масива.
  - На следващите R реда ще са числата от всеки от редовете в таблицата, C на брой, разделени с интервали.
- Входните данни винаги ще са валидни и в описания формат.

Изход

- Изходните данни трябва да бъдат отпечатани на конзолата.
- На R реда трябва да бъдат изведени числата от таблицата, сортирани по указаната колона.

Подсказки

- Ще се наложи да ползвате двумерен масив или масив от масиви.
- Тествайте задачата с различно големи масиви и различни алгоритми за сортиране.
- Обърнете внимание дали при стабилни и нестабилни алгоритми се получава еднакъв резултат.
- Коментирайте в клас има ли начин за минимизиране на размяната на редовете при сортирането.

Примери

Вход	Изход	Коментар	Вход	Изход	Коментар
3 4 **1** 1 2 3 4 3 1 2 4 2 3 1 2	1 2 3 4 2 3 1 2 3 1 2 4	Има 3 реда и 4 колони. Вторият и третият ред трябва да разменят местата си, ако сортираме по колона **1.**	4 2 **2** 1 2 3 1 2 3 4 4	3 1 1 2 2 3 4 4	Има 4 реда и 2 колони. Първият и вторият ред трябва да разменят местата си, ако сортираме по колона **2.**

Министерство на образованието и науката (МОН)

- Настоящият курс (презентации, примери, задачи, упражнения и др.) е разработен за нуждите на Национална програма "**Обучение за ИТ кариера**" на МОН за подготовка по професия "Приложен програмист".

- Курсът е базиран на учебно съдържание и методика, предоставени от **фондация "Софтуерен университет"** и се разпространява под **свободен лиценз CC-BY-NC-SA** (Creative Commons Attribution-Non-Commercial-Share-Alike 4.0 International).