

Упражнения: Сортиране чрез пряка селекция

Problem 1. Сортиране чрез пряка селекция

Сортирайте един масив от елементи с помощта на алгоритъма за пряка селекция (Selection 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 се облегнете на стола и му се насладете под формата на [цигански народен танц](#), представен от [AlgoRythmics](#). ☺

После създайте клас **SelectionSort** с един-единствен метод Sort

```
public class SelectionSort
{
    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. Алгоритъмът накратко е:

- Намираме **най-малката стойност** в масива
- **Разменяме я** с първата несортирана стойност (в началото това е първия елемент, после е втория и т.н.)
- **Това се повтаря** върху останалата част от масива (без сортираните стойности). Тоест:
 - търсим най-малката стойност от втория елемент до края и я поставяме на втора позиция
 - търсим най-малката стойност от третия елемент до края и я поставяме на трета позиция
 - и т.н., докато стигнем да сравняваме предпоследния и последния елемент

Алгоритъмът може да бъде реализиран с вложени цикли или с рекурсия:

- В първия случай външния цикъл ни служи да укажем от кой елемент да започнем да търсим минимална стойност и позицията, на която ще я поставим, а вътрешния цикъл е за намирането на минималната стойност чрез обхождане до края на масива.
- Във втория случай ще трябва да създадем допълнителен частен рекурсивен метод **Sort()**

```
private static void Sort(T[] arr, int lo, int hi)
```

В него директно реализираме алгоритъма:

- Ако е останал **само един елемент** в подмасива, то той **е вече сортиран** (което е и дъното на рекурсията).

```
if (lo >= hi)
{
    return;
}
```

- Ако не е, намираме **най-малката стойност** в масива (от позиция **lo** до **hi**)
- Разменяме я** с елемента на позиция **lo**
- Извикваме рекурсивно метода за останалата част от масива (след позиция **lo** до края)

Не забравяйте от публичния метод **Sort()** да извикате рекурсивния, като му подадете целия масив. ;-)

Problem 2. Визуализация на сортирането

Модифицирайте кода на предната задача, така че да служи за визуализация на метода на сортиране чрез пряка селекция. 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 отпечата масива след размяната.

Problem 3. Класиране на приема

От ръководството на вашето училище са ви помолили да участвате в комисията по приема на нови ученици. Ще ви предоставят лист с точките на всеки един ученик, кандидат за дадена паралелка, а вие трябва да направите програма, която ги извежда сортирани в ненарастващ ред, за да може комисията да определи кои ученици са приети в паралелката, кои са резерви и т.н.

Пример

Вход	Изход
95 98 103 109 48 92 25 106 160	160 109 106 103 98 95 92 48 25

Problem 4. Ученици по физическо

Вашият учител по физическо строява класовете винаги по височина. Можете ли да направите програма, която може да определи колко е височината на произволни двама от така подредените ученици при указване на поредния им номер в строя?

Вход

- Входните данни трябва да се прочетат от конзолата.
- На първия ред се подават неподредени цели числа, отделени едно от друго с интервал. Това са височините в сантиметри на всички ученици, преди да се подредят в строя.
- На втория ред се подават две цели числа, всяко в интервала от едно до броят на числата на първия ред. Това са позициите на тези ученици в подредения строй, на които искаме да разберем височините.
- Входните данни винаги ще са валидни и в описания формат. Не е необходимо да бъдат изрично проверявани.

Изход

- Изходните данни трябва да бъдат отпечатани на конзолата.
- На първия и единствен ред трябва да бъдат отпечатани две цели числа, разделени с интервал - височините на учениците, чиято позиция сме указали във входните данни.

Пример

Вход	Изход	Коментар
165 182 173 173 168 2 5	173 165	Класът, подреден по височина е 182 173 173 168 165 . Височините на вторият и петият ученик са удебелени .

Подсказки

Първо сортирайте масива и после изведете стойностите на елементите с указаните номера. Помислете може ли кодът да се оптимизира така, че да не се налага сортиране на целия масив. ;-)

Problem 5. Кросово бягане

Миналата седмица във вашия град бе проведено масово кросово бягане. Тъй като участниците са били наистина много, те са били разпределени да се състезават в два различни дни, по един и същ маршрут. На финаша е имало съдии, които са записали времето на пристигането на всеки състезател в секунди. На вас се пада честта да обедините тези два списъка в един и да оформите общата класация. Напишете програма, която да ви улесни в изпълнението на задачата.

Вход

- Входните данни трябва да се прочетат от конзолата.
- На първия ред ще има цели числа, подредени в ненамаляващ ред и отделени едно от друго с интервал. Това са времената на състезателите, участвали в кроса през първия ден.
- На втория ред ще са времената на състезателите, участвали в кроса през втория ден. Те са в същия формат, както числата от първия ред, но не е сигурно, че са същия брой.
- Входните данни винаги ще са валидни и в описания формат. Не е необходимо да бъдат изрично проверявани.

Изход

- Изходните данни (сортираният масив с класацията на всички състезатели, участвали в кроса, подредени в намаляващ ред) трябва да бъдат отпечатани на конзолата.

Пример

Вход	Изход	Коментар
956 989 1037 1095 948 992 1025 1062 1160	948 956 989 992 1025 1037 1062 1095 1160	За яснота участниците от първия ден са удебелени

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".



Министерство
на образованието
и науката



Национална
програма
„Обучение за
ИТ кариера“

- 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).



SoftUni
Foundation

