

Сортиране

Основно изискване към алгоритмите за сортиране е минималният разход на допълнителна памет. Друго изискване е минимален брой сравнения и размени на елементи.

По принцип сортирането се извършва чрез размяна на два елемента:

```
public void Swap(int[] arr, int i, int j) {  
    int temp = arr[i];  
    arr[i] = arr[j];  
    arr[j] = temp;  
}
```

Метод на пряката селекция (Selection sort)

1. Разделяме масива на сортирана и несортирана част. Сортираната част е с дължина i_1 , а несортираната е с дължина i_2 . Първоначално i_1 е с дължина 0, а $i_2 - n$
2. Намираме минималния елемент в несортираната част
3. Разменяме първия с минималния елемент
4. Увеличаваме размера на сортираната част с 1 и намаляваме размера на несортираната с 1
5. Продължаваме докато $i_2 > 1$
6. Край

```
public void MinIndex(int[] arr, int startFrom) {  
    int? minValue = null;  
    int minIndex = -1;  
    for(int i = startFrom; i < arr.Length; i++) {  
        if(arr[i] < minValue || minValue == null) {  
            minValue = arr[i];  
            minIndex = i;  
        }  
    }  
    return minIndex;  
}
```

```

public void SelectionSort(int[] arr) {
    int sortedSize = 0;
    int unsortedSize = arr.Length;
    while(unsortedSize > 1) {
        int min = MinIndex(arr);
        Swap(arr, sortedSize, min);
        sortedSize++;
        unsortedSize--;
    }
}

```

Сложност: $O(n^2)$

МЕТОД НА МЕХУРЧЕТО

1. Разделяме масива на сортирана и несортирана част (виж Selection sort)
2. Сравняваме два съседни елемента x_{j-1} и x_j в несортираната част на масива.

Първоначално x_j е последният елемент

3. Ако $x_{j-1} > x_j$ ги разменяме. $J--$. Ако $J > i_1 \rightarrow$ стъпка 2
4. i_1++ , i_2--
5. Продължаваме докато $i_2 > 1$

```

public void BubbleSort(int[] arr) {
    for(int i = 0; i < n-1; i++) {
        for(int j = n-1; j > i; j--) {
            if(arr[j-1] > arr[j]) {
                Swap(arr, j-1, j);
            }
        }
    }
}

```

Сложност: $O(n^2)$

СОРТИРАНЕ ЧРЕЗ ВМЪКВАНЕ

```

for (i = 1; i < N; i++) {
    x = arr[i];
    for(j = i; j > 0 && arr[j-1] > x; j++) {

```

```
    arr[j] == arr[j+1]
  }
  arr[j] = x;
}
```

Сложность: $O(n^2)$

Created with CosmicEveryday