

СAA – Упражнение 3

Бързо сортиране (quick sort)

Идеята на бързото сортиране е да се разменят елементи, които са максимално отдалечени един от друг – така се намалява броя на сравненията.

Принцип на действие:

1. Избира се "главен" или "водещ" елемент от списъка с елементи, които ще бъдат сортирани.
2. Списъкът се пренарежда така, че всички елементи, които са по-малки от "главния" се поставят вляво от него, а всички, които са по-големи - вдясно от него.
3. Рекурсивно се повтарят горните стъпки върху списъка с по-малките и списъка с по-големите елементи. Това разделяне продължава, докато размерът на частта, която се сортира стане 1.
4. Получените списъци се сливат (конкатенация) и се получава сортираният списък.

За илюстрация на действието на алгоритъма можете да разгледате файла `Sorting_quicksort_anim.gif`. В него е показано сортирането на елементите според водещ елемент, който е среден по стойност елемент за частта, която се сортира от масива.

В най-лошия случай алгоритъмът има сложност $O(n^2)$. В средния случай алгоритъмът има сложност $O(n \log n)$, такава е и минималната сложност. Най-тежкият случай е когато на всеки етап водещият елемент е минималният за сортирания участък и следователно той се разделя на една част с размер 1, а останалите елементи са в другата част.

Реализация на алгоритъма

```
void quicksort(int A[], int nch, int kr)
{
    //инициализация на необходимите променливи

    // определяне на медианата като среден по индекс елемент

    m=A[(nch+kr)/2];

    do
    {
        // търсене на елемент, по-голям от медианата, обхождането на масива е от
        // началото към края му

        while (A[i]<m)
            i++;
```

```

// търсене на елемент, по-малък от медианата, обхождането на масива е от
// края към началото

while (A[j]>m)
    j--;

// ако двата показалеца не са се разминали се прави размяна на
// елементите A[i] и A[j]

if(i<=j)
{
    размяна на елементите A[i] и A[j]
    i++;
    j--;
}
}
while (i<=j);

// разделяне на масива на два подмасива – от началото до j-та позиция и от i-та
// позиция до края – за тези два подмасива се извиква рекурсивно функцията
// quicksort

if (nch<j)
    quicksort(A,nch,j);
if (i<kr)
    quicksort(A,i,kr);
}

```

Задачи:

1. Реализирайте алгоритъма за бързото сортиране с отделна функция. В main функцията въведете елементите на масива, който ще бъде сортиран, извикайте функцията за бързо сортиране, след което разпечатайте елементите на сортирания вече масив. Масивът, който ще бъде сортиран да съдържа реални числа (тип float).
2. При така описаната реализация определете броя на рекурсивните обръщения във функцията quicksort в ситуацията, при която:
 - А) водещият елемент на всяка стъпка е минималният за сортирания участък.
 - Б) водещият елемент на всяка стъпка е максималният за сортирания участък.
 Обосновете отговорите си.