

Методът на мехурчето (на английски: Bubble sort) е прост сортиращ алгоритъм. Алгоритъмът започва в началото на сортирания се списък. Той сравнява първия и втория елемент, и ако първият е по-голям от втория, ги разменя. Продължава да прави с това с всички съседни двойки елементи до края на сортирания се списък. След това повтаря същото действие още толкова пъти, докато накрая при обхождането на целия списък не е направено нито една размяна на два съседни елемента. Средният и най-лош случай на този алгоритъм е $O(n^2)$ и не се използва за сортиране на големи неподредени множества от данни. Методът на мехурчето може да се използва за малки множества или за множества, където има елементи близо до очакваното си място. Например, ако някой елемент не е на мястото си, на разстояние един елемент (например 112 и 111), методът на мехурчето ще обходи един път множеството и ще направи размяната, а на второто обхождане няма да направи размяна и ще завърши сортирането, и това ще отнеме само $2n$.

Алгоритъмът за сортиране чрез пряка селекция (на английски: Selection sort) е неефективен със изчислителна сложност $O(n^2)$. Подобен алгоритъм, който има по-добра производителност, е алгоритъмът за сортиране чрез вмъкване. Сортирането чрез пряка селекция впечатлява с простотата си, а също така в дадени ситуации има предимства пред някои сложни алгоритми. Алгоритъмът намира най-малкия елемент в списъка и го разменя с първия елемент. След това търси втория най-малък и го поставя на второ място. Действието се повтаря, докато се стигне до края на списъка. При този алгоритъм не се правят повече от n размени. Този алгоритъм е полезен в случаи, когато размяната на елементи е тежка операция и трябва да се минимизира.

Сортирането чрез броене е алгоритъм за сортиране на група от обекти спрямо техните ключове, които са малки цели числа; това е алгоритъм за сортиране на цели числа. Алгоритъмът се изпълнява, като брой обектите с различни стойности на ключа, и използвайки аритметика за тези числа за определяне на позициите на всяка ключова стойност в изходната поредица. Времето за изпълнение е линейно в рамките на броя на елементите и разликата между максималната и минималната ключова стойност, така че е подходящ само за директна употреба в случаите, когато разликата в ключовете не е значително по-голяма от броя на елементите. Най-общо казано, алгоритъмът обхожда елементите, изчислявайки хистограма на броя пъти, в които даден ключ се среща във входните данни. След това се изчислява префикс сума (второ обхождане на обхвата на възможните ключове), за да се определи за всеки началната позиция в изходния масив от елементи със съответния ключ. Накрая елементите се обхождат отново, като се преместват на сортираните позиции в изходния масив.

Radix sort е алгоритъм, който може да се използва за сортиране на елементи във лексикографски ред. Елементите могат да са символни низове или числа. Сортирането на елементите започва съобразно с цифрата, която има най-малко значение за големината на числото (пр. най-лявата, която показва броят на единиците) и продължава през по-значещите цифри (тези за десетици, стотици и т.н.) Този тип алгоритъм има сложност $O(nk)$, където n е броя на сортираните елементи, а k е средният брой на техните цифри. Radix sort е възникнал като алтернатива на други алгоритми за сортиране с голямо бързодействие (като heap sort и merge sort), които имат сложност $O(n \log n)$. Тези алгоритми имат по-голяма сложност, но могат да сортират елементите по методи, по-различни от лексикографския. Този алгоритъм разпределя елементите на групи спрямо стойността на най-дясната тяхна цифра (най-малко значещата лексикографски). Във всяка група елементите са подредени във първоначалния ред. Броя на групите е еднакъв с броя на стойностите, които може да приема всяка цифра. След това този процес се повтаря на всяка съседна цифра, която има по-голямо

лексикографско значение, докато не останат повече цифри.

Вземи най-малко значещата цифра на всеки елемент

Групирай елементите спрямо стойността на тази цифра, като запазиш реда им по групи

Повтори този процес със по-значещите цифри

Подреждането в стъпка 2 обикновено се прави чрез bucket sort или counting sort които са ефективни в случая ако има малко елементи за сравнение.

Съществува и друга вариация на този алгоритъм за сортиране, която сравнява елементите първо по тяхната най-значеща цифра и след това преминава към по-малко значещите.