

Көпіршікті сұрыптау - ең танымал сұрыптау алгоритмдерінің бірі. Мұнда көрші элементтердің мәндерін дәйекті түрде салыстыру керек және алдыңғысы келесіден үлкен болса, сандарды ауыстыру керек. Осылайша, үлкен мәндері бар элементтер тізімнің соңында, ал кіші мәндері бар элементтер басында қалады.

Сұрыптаудың бұл түрі информатика пәнімен танысудың басында оқытылады, өйткені ол сұрыптау түсінігін барынша қарапайым түрде көрсетеді.

Бұл тәсіл тізімді қайталайды және көрші элементтерді салыстырады. Егер тапсырыс дұрыс болмаса, олар ауыстырылады. Бұл барлық элементтер дұрыс тәртіпте болғанша жалғасады. Көпіршікті сұрыптаудағы қайталаулар санының көп болуына байланысты оның ең нашар күрделілігі $O(n^2)$ болып табылады.

Python:

```
def bubble_sort(arr):
    def swap(i, j):
        arr[i], arr[j] = arr[j], arr[i]
    n = len(arr)
    swapped = True
    x = -1
    While swapped:
        swapped = False
        x = x + 1
        for i in range(1, n-x):
            if arr[i - 1] > arr[i]:
                swap(i - 1, i)
```

swapped = True

Тандау сұрыптауы іске асыру үшін ең оңай сұрыптау алгоритмі болуы мүмкін. Басқа ұқсас алгоритмдердің көпшілігі сияқты, ол салыстыру операциясына негізделген. Әрбір элементті әрқайсысымен салыстыру және қажет болған жағдайда алмасуды жүзеге асыру әдіс ретті қажетті реттелген пішінге әкеледі.

Алгоритм идеясы өте қарапайым. N өлшемді A массиві болсын, онда іріктеу сұрыптауы келесіге дейін азаяды:

1] $A[i]$ тізбегінің бірінші элементін алайық, мұндағы i - элемент нөмірі, бірінші i үшін ол 1-ге тең;

2] тізбектің минималды (максималды) элементін табу және оның нөмірін кілттік айнымалыға сақтау;

3] егер бірінші элементтің нөмірі мен табылған элементтің нөмірі сәйкес келмесе, яғни егер кілт?1 болса, онда бұл екі элемент мәндермен алмасады, әйтпесе ешқандай манипуляциялар болмайды;

4] i -ді 1-ге көбейтіңіз және массивтің қалған бөлігін сұрыптауды жалғастырыңыз, атап айтқанда 2 саны бар элементтен N -ге дейін, өйткені $A[1]$ элементі өз орнын алып қойған;

Әрбір келесі қадаммен алгоритм жұмыс істейтін ішкі жиымның өлшемі 1-ге азаяды, бірақ бұл сұрыптау әдісіне әсер етпейді, ол әрбір қадам үшін бірдей.

Ең нашар уақыт

$O(n^2)$

Ең жақсы уақыт

$O(n^2)$

Орташа уақыт

$O(n^2)$

Жад құны

$O(n)$ барлығы, $O(1)$ қосымша

Python:

```
def swap(arr, i, j):
    arr[i], arr[j] = arr[j], arr[i]

def select_sort(arr):
    i = len(arr)
    while i > 1:
        max = 0
        for j in xrange(i):
            if arr[j] > arr[max]:
                max = j
        swap(arr, i - 1, max)
        i -= 1
```

Қарапайым $\Theta(n^2)$ орташа регистрлік алгоритмдердің ішінде таңдау сұрыптауы әрқашан дерлік көпіршікті сұрыптау мен гномды сұрыптаудан асып түседі, бірақ әдетте кірістіру сұрыптауынан асып түседі. Кірістіру сұрыптауы өте ұқсас, себебі k -итерациядан кейін массивтегі бірінші k элементтер сұрыпталған тәртіпте болады. Кірістіру сұрыптасының артықшылығы мынада, ол $k + 1$ -ші элементті орналастыру үшін қажет болғанша көп элементтерді сканерлейді, ал таңдау сұрыптауы $k + 1$ -ші элементті табу үшін барлық басқа элементтерді сканерлеуі керек.