

الگوریتم مرتب سازی انتخابی به طور ذاتی ناپایدار است این به این معنی است که اگر دو عنصر برابر در ارایه وجود داشته باشد ترتیب نسبی آنها بعد از مرتب سازی ممکن است تغییر کند. مثلاً فرض کنید این ارایه را داریم $[4a, 2, 4b, 1]$ در اینجا دو عنصر 4 داریم. با استفاده از الگوریتم مرتب سازی انتخابی مراحل به صورت زیر است

1. کوچکترین عنصر 1 پیدا میشود و با $4a$ تعویض میشود

2. الان $2, 4a, 4b$ بررسی میشود و 2 باقی میماند

3. در این مرحله $4a, 4b$ مقایسه میشود

در نهایت ترتیب $4a, 4b$ تغییر کرده و الگوریتم ناپایدار است

برای پایدار کردن الگوریتم، می توانیم در هنگام انتخاب حداقل (یا حداکثر) عنصر، از ترتیب اولیه آنها استفاده کنیم. یکی از راههای ساده برای این کار، این است که به هر عنصر یک ایندکس اضافه کنیم و سپس براساس مقدار و ایندکس مرتب کنیم.

```

def stable_selection_sort(arr):
    indexed_arr = [(value, index) for index, value in
enumerate(arr)]
    n = len(indexed_arr)
    for i in range(n):
        min_index = i
        for j in range(i + 1, n):
            if indexed_arr[j][0] < indexed_arr[min_index][0] or \
                (indexed_arr[j][0] == indexed_arr[min_index][0] and
indexed_arr[j][1] < indexed_arr[min_index][1]):
                min_index = j
            indexed_arr[i], indexed_arr[min_index] =
indexed_arr[min_index], indexed_arr[i]
    return [value for value, index in indexed_arr]

arr = [4, 3, 4, 1]
sorted_arr = stable_selection_sort(arr)
print(sorted_arr)

```

خروجی: [1,3,4,4]