

Sprawozdanie

1. Teoria

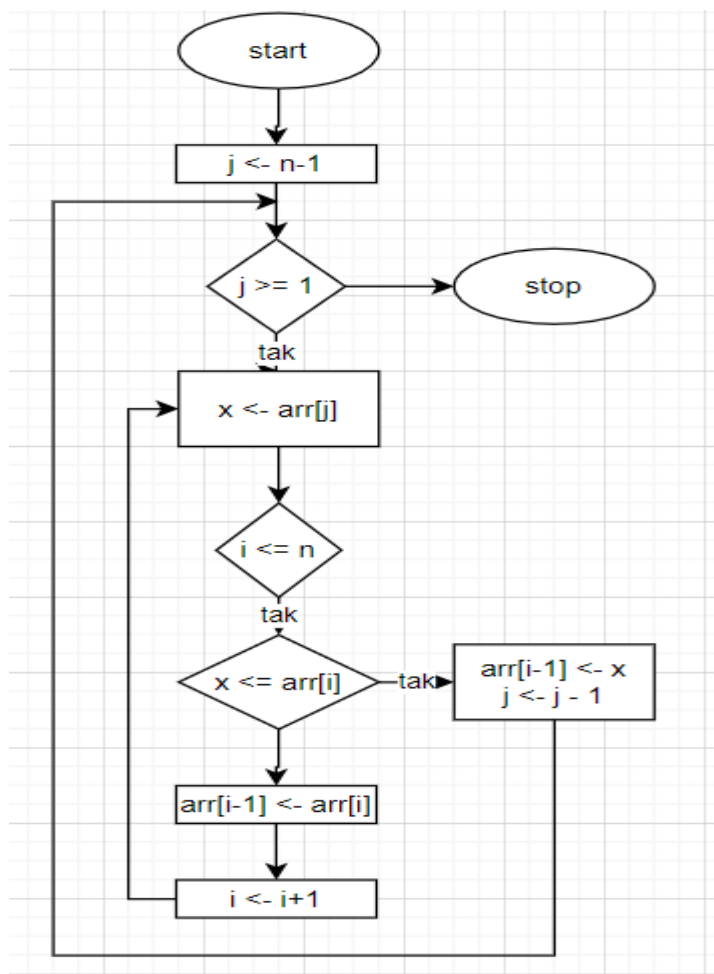
Sortowanie przez wstawianie -jeden z najprostszych algorytmów sortowania, którego zasada działania odzwierciedla sposób w jaki ludzie ustawiają karty – kolejne elementy wejściowe są ustawiane na odpowiednie miejsca docelowe. Jest efektywny dla niewielkiej liczby elementów, jego złożoność wynosi $O(n^2)$. Pomimo tego, że jest znacznie mniej wydajny od algorytmów takich jak quicksort czy heapsort, posiada pewne zalety:

- ⑩ liczba wykonanych porównań jest zależna od liczby inwersji w permutacji, dlatego algorytm jest wydajny dla danych wstępnie posortowanych,
- ⑩ jest wydajny dla zbiorów o niewielkiej liczebności,
- ⑩ jest stabilny.

Istnieje modyfikacja algorytmu, pozwalająca zmniejszyć liczbę porównań. Zamiast za każdym razem iterować po już posortowanym fragmencie (etap wstawiania elementu), można posłużyć się wyszukiwaniem binarnym. Pozwala to zmniejszyć liczbę porównań do $O(n \log n)$, nie zmienia się jednak złożoność algorytmu, ponieważ liczba przesunięć elementów to nadal $O(n^2)$.

Sortowanie przez kopcowanie -zwane również sortowaniem stogowym – jeden z algorytmów sortowania, choć niestabilny, to jednak szybki i niepochłaniający wiele pamięci (złożoność czasowa wynosi $O(n \log n)$ a pamięciowa -przy czym jest to rozmiar sortowanych danych, złożoność pamięciowa dodatkowych struktur wynosi $O(1)$ jest to zatem algorytm sortowania *w miejscu*). Jest on w praktyce z reguły nieco wolniejszy od sortowania szybkiego, lecz ma lepszą pesymistyczną złożoność czasową (przez co jest odporny np. na atak za pomocą celowo spreparowanych danych, które spowodowałyby jego znacznie wolniejsze działanie).

2. Schematy blokowe



Sortowanie przez wstawianie.

Sortowanie przez kopcowanie.

