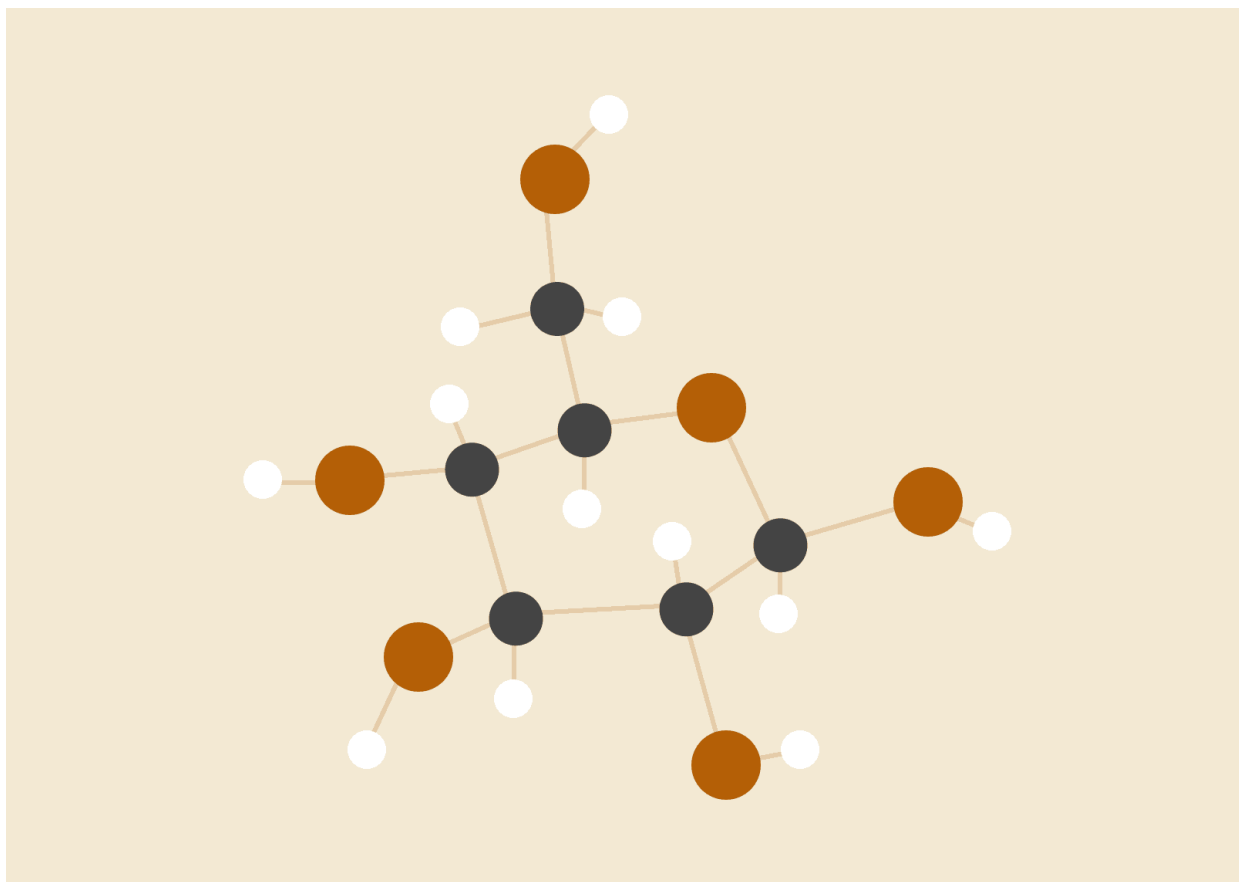


ЛАБОРАТОРНА РОБОТА №1.1

Алгоритми та структури даних



Денісов Вадим Ігорович

2.20.2022

ІК-9-19Б1ПЗ

ТЕМА

Дослідження принципів роботи алгоритмів сортування.

МЕТА

Ознайомитися з різновидами алгоритмів сортування. Вивчити алгоритм сортування вставками, методом бульбашки і їх різновиди, алгоритм швидкого сортування та сортування злиттям. Вивчити принцип проведення аналізу обчислювальної складності алгоритмів.

ЗАВДАННЯ

1. Вивчити основні алгоритми сортування та принципи аналізу обчислювальної складності алгоритмів.
2. Реалізувати алгоритм відповідно до варіанту.
3. Розрахувати обчислювальну ефективність алгоритму. Провести порівняльний аналіз з модифікованими алгоритмами того ж принципу сортування.

ТЕОРЕТИЧНІ ВІДОМОСТІ

1. Алгоритм сортування – це алгоритм для впорядкування елементів у масиві. У випадку, коли елемент масиву має кілька полів, поле, що служить критерієм порядку, називається ключем сортування. На практиці як ключ часто виступає число, а в інших полях зберігаються будь-які дані, що ніяк не впливають на роботу алгоритму.
2. Алгоритм сортування вставкою - на кожному кроці алгоритму ми вибираємо один з елементів вхідних даних і вставляємо його на потрібну позицію у вже відсортованому списку доти, доки набір вхідних даних не буде вичерпано.

3. Алгоритм сортування Шелла - На початку обираються m -елементів: d_1, d_2, \dots, d_m , причому, $d_1 > d_2 > \dots > d_m = 1$. Потім виконується m впорядкувань методом включення, спочатку для елементів, що стоять через d_1 , потім для елементів через d_2 і т.д. до $d_m = 1$. Ефективність досягається тим, що кожне наступне впорядкування вимагає меншої кількості перестановок, оскільки деякі елементи вже стали на свої місця.

АЛГОРИТМ (мовою JavaScript)

1. Код алгоритму сортування вставками.

```
131   const insertionSort = arr => {
132
133     for (let i = 1, l = arr.length; i < l; i++) {
134       const current = arr[i];
135       let j = i;
136
137       while (j > 0 && arr[j - 1] > current) {
138         arr[j] = arr[j - 1];
139         j--;
140       }
141       arr[j] = current;
142     }
143     return arr;
144   };
145
```

2. Код алгоритму сортування Шелла.

```
146   function shellSort(arr) {
147     let n = arr.length;
148
149     for (let gap = Math.floor(n/2); gap > 0; gap = Math.floor(gap/2)) {
150
151       for (let i = gap; i < n; i++) {
152         let temp = arr[i];
153
154         let j;
155         for (j = i; j >= gap && arr[j-gap] > temp; j-=gap) {
156           arr[j] = arr[j-gap];
157         }
158
159         arr[j] = temp;
160       }
161
162       return arr;
163     }
  }
```

3. Код що спрацюває після натиску кнопки 'Start!'

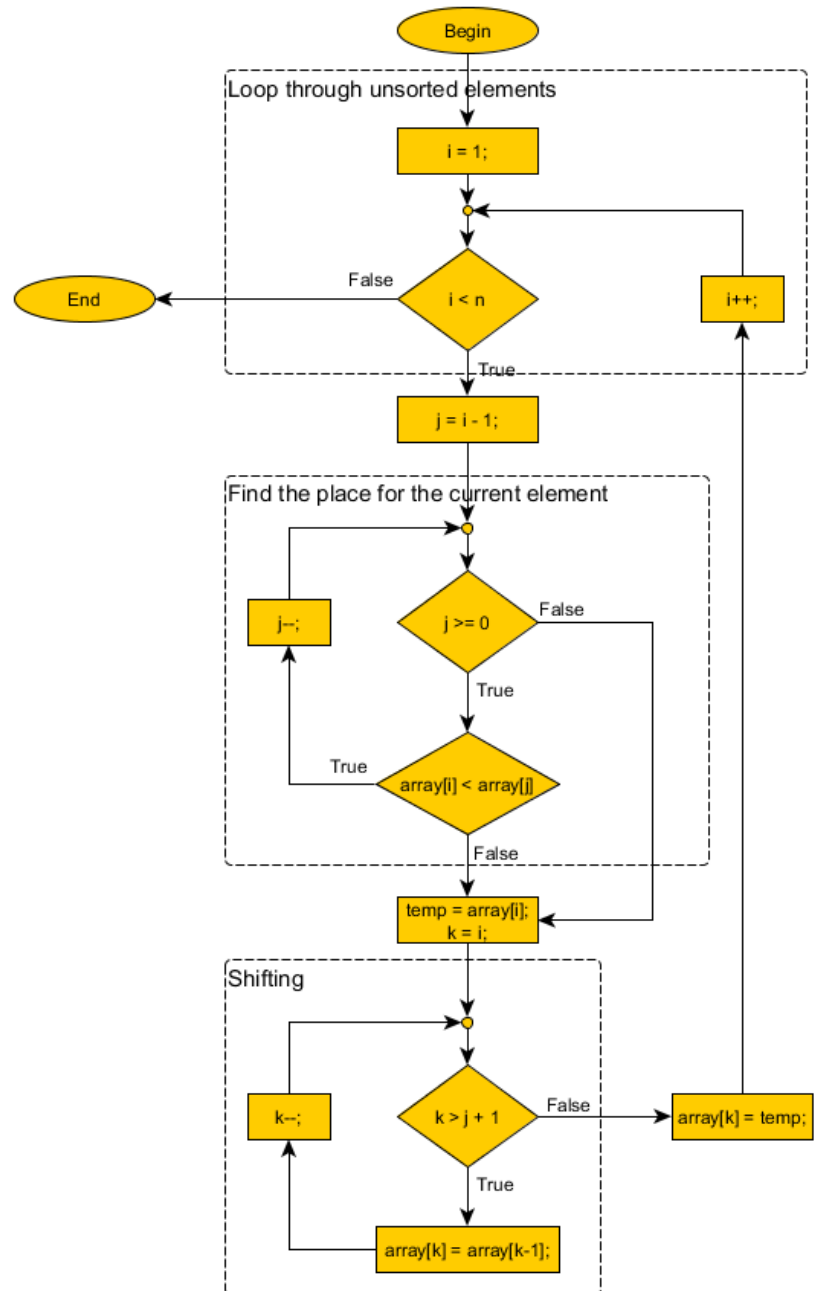
```
181 button.addEventListener('click', event => {
182     const n = document.querySelector('.container_input').value;
183     randArr = [];
184
185     errorDiv.style.display = 'none';
186     numCheck(n);
187
188     if (check) {
189         arrCreate(n)
190         randArr2 = copy(randArr);
191
192         console.log(randArr);
193
194         startTime = performance.now();
195         console.log(insertionSort(randArr));
196         endTime = performance.now();
197
198         resultDiv1.textContent = `InsertionSort completed in ${endTime - startTime} milliseconds`;
199
200         console.log(randArr2);
201
202         startTime = performance.now();
203         console.log(shellSort(randArr2));
204         endTime = performance.now();
205         resultDiv2.textContent = `ShellSort completed in ${endTime - startTime} milliseconds`;
206     }
207 }
208 })
```

РЕЗУЛЬТАТИ

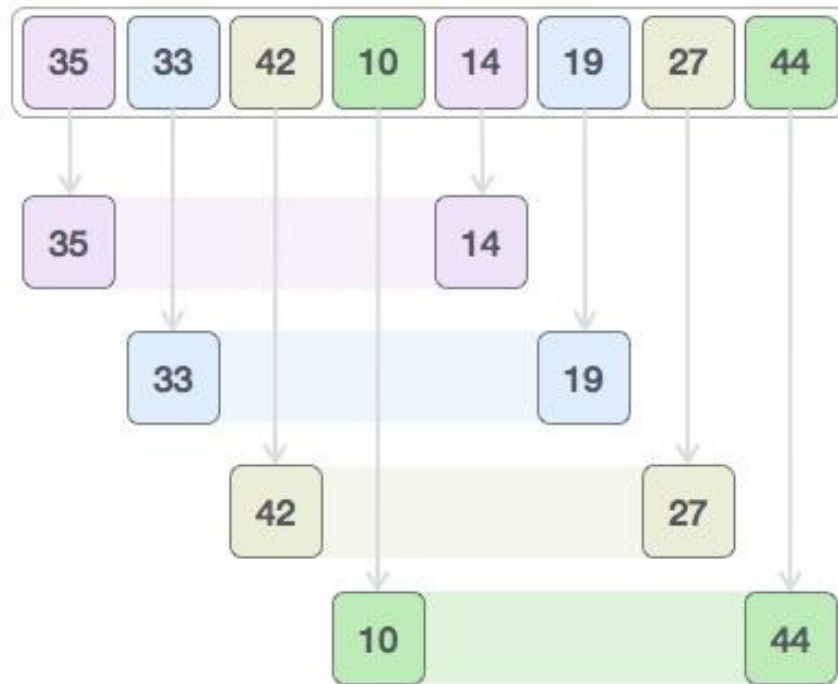
Алгоритми / Обсяг масиву / Витрати часу	100	1.000	10.000	100.000
Сортування вставкою	0 - 1 мс	1 - 3 мс	70 - 80 мс	7025 - 7450 мс
Сортування Шелла	0 - 1 мс	0 - 1 мс	3 - 6 мс	29 - 35 мс

БЛОК-СХЕМИ

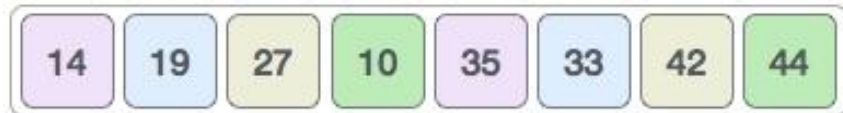
1. Метод вставки.



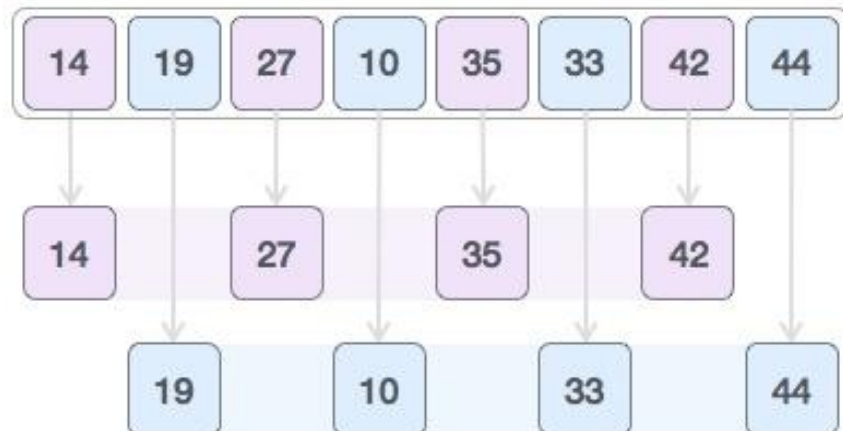
2. Метод Шелла



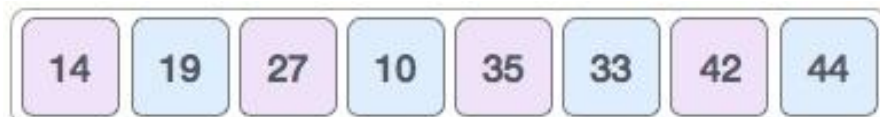
a.



b.



c.



d.



ВИКОРИСТАНІ ДЖЕРЕЛА

1. https://vue.gov.ua/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D1%81%D0%BE%D1%80%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F
2. <https://web.archive.org/web/20070601160908/http://ai-blog.org/node/ExchangeSort>
3. <http://www.unicyb.kiev.ua/Library/PROG/Zmist.htm>