

सॉर्टिंग

पाठ्यपुस्तक के प्रश्न

वस्तुनिष्ठ प्रश्न

प्रश्न 1. बबल एल्गोरिथ्म की जटिलता है

- (अ) $O(N)$
- (ब) $O(N^2)$
- (स) $O(\log N)$
- (द) $O(N \log N)$

उत्तर: (ब) $O(N^2)$

प्रश्न 2. मर्ज एल्गोरिथ्म की जटिलता है

- (अ) $O(N)$
- (ब) $O(N^2)$
- (स) $O(\log N)$
- (द) $O(N \log N)$

उत्तर: (द) $O(N \log N)$

प्रश्न 3. चयन एल्गोरिथ्म की जटिलता है

- (अ) $O(N)$
- (ब) $O(N^2)$
- (स) $O(\log N)$
- (द) $O(N \log N)$

उत्तर: (ब) $O(N^2)$

प्रश्न 4. कौन सा अच्छा सॉर्टिंग एल्गोरिथ्म है?

- (अ) चयन सॉर्टिंग
- (ब) निवेशन सॉर्टिंग
- (स) त्वरित सॉर्टिंग
- (द) कोई नहीं

उत्तर: (स) त्वरित सॉर्टिंग

प्रश्न 5. त्वरित क्रमबद्ध एल्गोरिथ्म की जटिलता है।

- (अ) $O(N)$
- (ब) $O(\log N)$
- (स) $O(N^2)$
- (द) $O(N \log N)$

उत्तर: (द) $O(N \log N)$

लघु उत्तरीय प्रश्न

प्रश्न 1. सॉर्टिंग क्या है?

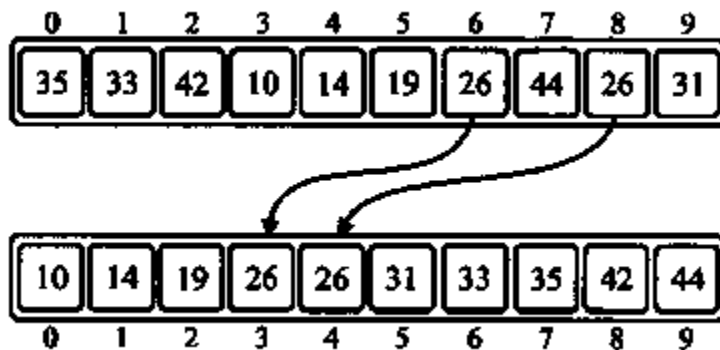
उत्तर- सॉर्टिंग (Sorting) एक विशेष स्वरूप में डेटा को व्यवस्थित करने को संदर्भित करता है। सॉर्टिंग एल्गोरिथ्म डेटा को एक विशेष क्रम में व्यवस्थित करने का तरीका निर्दिष्ट करती है। सबसे आम क्रम संख्यात्मक या वर्णानुक्रम हैं। यदि डेटा एक क्रमबद्ध तरीके से संग्रहित किया गया है तो सॉर्टिंग का सर्वाधिक महत्व डाटा सर्च को आसान बनाने में है। सॉर्टिंग डेटा को और अधिक पठनीय प्रारूप में प्रदर्शित करने के लिए भी प्रयोग की जाती है। वास्तविक जीवन में सॉर्टिंग के कुछ उदाहरण हैं :

टेलीफोन निर्देशिका – टेलीफोन निर्देशिका, लोगों के टेलीफोन नम्बरों को उनके नाम के अनुसार क्रमबद्ध करके संग्रहीत करती है जिससे नामों को आसानी से सर्च किया जा सकता है।

शब्दकोश – शब्दकोश में शब्द वर्णमाला के क्रम से संग्रहीत किये जाते हैं इसलिए किसी भी शब्द को सर्च करना आसान हो जाता है।

प्रश्न 2. स्थिर सॉर्टिंग क्या है?

उत्तर- स्टेबल सॉर्टिंग (Stable Sorting) – सॉर्टिंग एल्गोरिथ्म, तत्वों को सॉर्ट करने के बाद, एक जैसे तत्वों के क्रम जिसमें वो प्रकट होते हैं को परिवर्तित नहीं करती है उनको स्टेबल सॉर्टिंग कहा जाता है।



प्रश्न 3. इन-प्लेस सॉर्टिंग क्या है?

उत्तर- इन-प्लेस सॉर्टिंग (In-place Sorting)-सॉर्टिंग एल्गोरिथ्म को तुलना और कुछ डेटा तत्वों के अस्थायी भण्डारण के लिए कुछ अतिरिक्त स्थान की आवश्यकता हो सकती है। इन-प्लेस सॉर्टिंग एल्गोरिथ्म को किसी भी अतिरिक्त जगह की आवश्यकता नहीं होती है और इसलिए इन्हें सॉर्टिंग इन-प्लेस कहा जाता है, उदाहरण के लिए, ऐरे के भीतर ही सॉर्टिंग। बबल सॉर्ट इन-प्लेस सॉर्टिंग का एक उदाहरण है।

प्रश्न 4. त्वरित एल्गोरिथ्म के लिए सबसे खराब स्थिति (Worst case) का रन टाइम, क्या है?

उत्तर- त्वरित (Quick) सॉर्ट एल्गोरिथ्म के लिए सबसे खराब स्थिति का रन टाइम (n^2) होता है।

प्रश्न 5. त्वरित एल्गोरिथ्म के लिए सबसे खराब स्थिति (Worst case) क्या है?

उत्तर- त्वरित एल्गोरिथ्म के लिए सबसे खराब स्थिति निम्न केसों में होती है

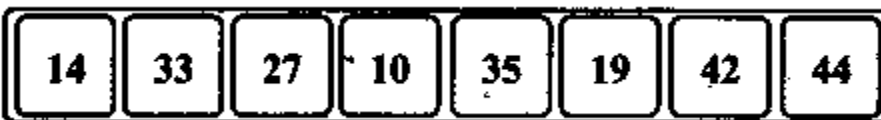
- जब ऐरे पहले से ही सॉर्टेड हो।
- जब ऐरे पहले से ही उल्टे क्रम में सॉर्टेड हो।
- जब सारे तत्व समान हों।

निबंधात्मक प्रश्न

प्रश्न 1. मर्ज सॉर्टिंग विस्तार में समझाइए।

उत्तर- मर्ज (Merge) सॉर्टिंग : मर्ज सॉर्ट डिवाइड (विभाजित) एण्ड कॉन्कर (जीत) पर आधारित एक सॉर्टिंग तकनीक है। इसकी सबसे खराब मामले में (worst-case) कॉम्प्लेक्सिटी $O(n \log n)$ होने के कारण यह सबसे अच्छी एल्गोरिथ्म में से एक है। मर्ज सॉर्ट ऐरे को दो बराबर हिस्सों में तोड़ती है और फिर उन्हें एक क्रमबद्ध ढंग से जोड़ती है।

मर्ज सॉर्ट को समझने के लिए हम एक निम्नलिखित अवर्गीकृत ऐरे लेते हैं।



हम जानते हैं कि मर्ज सॉर्ट पहले पूरी ऐरे को पुनरावृत्तीय तरीके से बराबर हिस्सों में बाँटती है जब तक कि परमाणु (atomic) या अविभाज्य मान प्राप्त नहीं हो जाते हैं। हम यहाँ देखते हैं कि 8 मानों की एक ऐरे 4 आकार की दो ऐरे में बाँट गयी है।



यह मूल मानों की उपस्थिति के अनुक्रम को नहीं बदलता है। अब हम इन दो ऐरे को हिस्सों में विभाजित करते हैं।



हम आगे इन ऐरे को और विभाजित करते हैं और हमें परमाणु मान प्राप्त होते हैं जिनको और अधिक विभाजित नहीं किया जा सकता है।



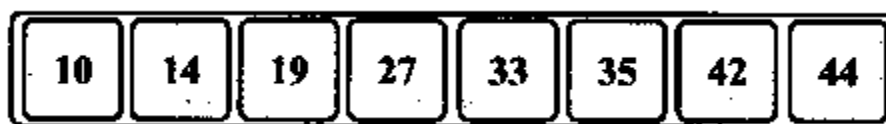
अब, हम उन्हें ठीक उसी तरीके से सम्मिलित करते हैं जैसे उन्हें तोड़ा था। हम पहले प्रत्येक लिस्ट के तत्व की तुलना करते हैं और फिर एक क्रमबद्ध ढंग से उन्हें एक दूसरी लिस्ट में सम्मिलित करते हैं। हम जानते हैं कि 14 और 33 सॉर्टेड स्थिति में ही हैं। हम 27 और 10 की तुलना करते हैं और 2 मानों की लक्ष्य लिस्ट में हम पहले 10 को डालते हैं और उसके पीछे 27 को। हम 19 और 35 का क्रम बदलते हैं जबकि 42 और 44 को क्रमिक रूप से रखा जाता है।



संयोजन चरण के अगले पुनरावृत्ति में, हम दो डेटा मानों की लिस्ट की तुलना करते हैं, और उन्हें एक सॉर्टेड क्रम में डेटा मानों की लिस्ट में मर्ज कर देते हैं।

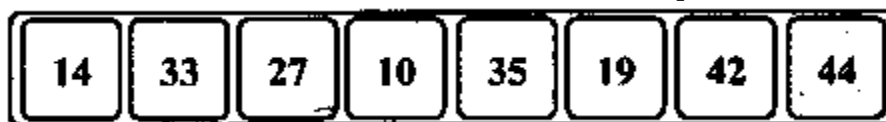


अंतिम विलय के बाद, लिस्ट इस तरह दिखेगी



प्रश्न 2. कौन-सी सबसे अच्छी सॉर्टिंग एल्गोरिथ्म है और क्यों?

उत्तर- मर्ज सॉर्टिंग सबसे अच्छी सॉर्टिंग एल्गोरिथ्म है। इसकी सबसे खराब मामले में (worst-case) कॉम्प्लेक्सिटी $O(n \log n)$ होने के कारण यह सबसे अच्छी एल्गोरिथ्म में से एक है। मर्ज सॉर्ट डिवाइड (विभाजित) एण्ड कॉन्कर (जीत) पर आधारित एक सॉर्टिंग तकनीक है। मर्ज सॉर्ट को समझने के लिए हम एक निम्नलिखित अवर्गीकृत ऐरे लेते हैं।



हम जानते हैं कि मर्ज सॉर्ट पहले पूरी ऐरे को पुनरावृत्तीय तरीके से बराबर हिस्सों में बांटती है जब तक कि परमाणु (atomic) या अविभाज्य मान प्राप्त नहीं हो जाते हैं। हम यहाँ देखते हैं कि 8 मानों की एक ऐरे 4

आकार की दो ऐरे में बंट गयी है।



यह मूल मानों की उपस्थिति के अनुक्रम को बदलता है। अब हम इन दो ऐरे को हिस्सों में विभाजित करते हैं।



हम आगे इन ऐरे को और विभाजित करते हैं इमें परमाणु मान प्राप्त होते हैं जिनको और अधिक विभाजित नहीं किया जा सकता।



अब, हम उन्हें ठीक उसी तरीके से सम्मिलित करते हैं जैसे उन्हें तोड़ा था।

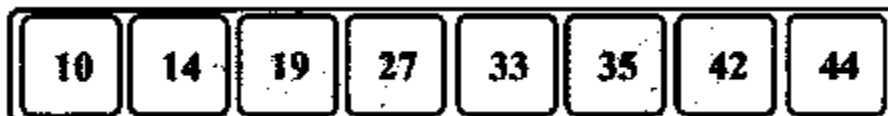
हम पहले प्रत्येक लिस्ट के तत्व की तुलना करते हैं और फिर एक क्रमबद्ध ढंग से उन्हें एक दूसरी लिस्ट में सम्मिलित करते हैं। हम जानते हैं कि 14 और 33 सॉर्टेड स्थिति में ही हैं। हम 27 और 10 की तुलना करते हैं और 2 मानों की लक्ष्य लिस्ट में हम पहले 10 को डालते हैं और उसके पीछे 27 को। हम 19 और 35 का क्रम बदलते हैं जबकि 42 और 44 को क्रमिक रूप से रखा जाता है।



संयोजन चरण के अगले पुनरावृत्ति में, हम दो डेटा मानों की लिस्ट की तुलना करते हैं, और उन्होंने एक सॉर्टेड क्रम में डेटा मानों की लिस्ट में मर्ज कर देते हैं।



अंतिम विलय के बाद, लिस्ट इस तरह दिखेगी



Algorithm : मर्ज सॉर्ट लिस्ट को पुनरावृत्तीय तरीके से बराबर हिस्सों में बांटती है जब तक कि उसे ओर अधिक विभाजित नहीं किया जा सकता। परिभाषा के अनुसार, अगर लिस्ट में केवल एक ही तत्व है, तो यह लिस्ट सॉर्टेड है। फिर, मर्ज सॉर्ट छोटी सॉर्टेड सूचियों को इस तरह सम्मिलित करती है ताकि नयी बनाने वाली सूची भी सॉर्टेड ही रहे।

Step 1 – अगर लिस्ट में केवल एक ही तत्व है, तो यह लिस्ट सॉर्टेड है।

Step 2 – लिस्ट को पुनरावृत्तीय तरीके से दो बराबर हिस्सों में बांटना जब तक कि उसे और अधिक विभाजित नहीं किया जा सकता।

प्रश्न 3. त्वरित सॉर्टिंग समझाइए।

उत्तर- त्वरित (Quick) सॉर्टिंग: त्वरित एक प्रकार की अत्यन्त कुशल सॉर्टिंग एल्गोरिथ्म है और डेटा के ऐरे को छोटे ऐरे में विभाजन करने पर आधारित है। एक बड़ा ऐरे दो ऐरे में विभाजित किया जाता है, जिनमें से एक ऐरे में निर्धारित मान (जिसके आधार पर विभाजन किया जाता है और इसे पाइवोट कहते हैं) की तुलना में छोटे मान रखता है, और दूसरे ऐरे में पाइवोट मान से अधिक मानों को रखा जाता है।

त्वरित सॉर्टिंग एक ऐरे को विभाजित करती है और उसके बाद दो परिणामस्वरूप सब-ऐरे को सॉर्ट करने के लिए खुद को बारी-बारी से दो बार कॉल करती है। यह एल्गोरिथ्म बड़े आकार के डेटा सेट के लिए काफी कुशल है।

इस एल्गोरिथ्म की औसत (average) और सबसे खराब (worst case) स्थिति में कॉम्प्लेक्सिटी $O(n \log n)$ है, जहाँ n सॉर्ट किये जाने वाले तत्वों की संख्या है।

त्वरित सॉर्टिंग में विभाजन : निम्नलिखित उदाहरण यह बताता है कि कैसे एक ऐरे में पाइवोट मान को सर्च किया जाता है।

पाइवोट मान लिस्ट को दो भागों में बाँटता है और बारी-बारी से, हम प्रत्येक उप-सूचियों के लिए पाइवोट मान का पता लगाते हैं जब तक कि सभी सूचियों केवल एक ही तत्व नहीं रह जाता।

A[6]	A[7]	A[8]	A[9]	A[10]	A[11]	A[12]
(98)	84	65	108	60	96	(72)
72	84	65	(108)	60	96	(98)
72	84	65	(98)	60	(96)	108
72	84	65	96	60	(98)	108

प्रश्न 4. Selection और Insertion सॉर्टिंग के बीच अन्तर बताइए।

उत्तर- चयन (Selection) सॉर्टिंग

यह एक सरल सॉर्टिंग एल्गोरिथ्म है। यह एक इन-प्लेस तुलना-आधारित सॉर्टिंग एल्गोरिथ्म है इसमें सूची दो भागों में विभाजित होती है, सॉर्ट किया हुआ भाग बाईं ओर तथा सॉर्ट न किया हुआ भाग दायी ओर रहता है। शुरू में, सॉर्ट किया गया हुआ भाग खाली रहता है और सम्पूर्ण सूची सॉर्ट न किये हुए भाग में होती है। अवर्गीकृत (अनसॉर्टेड) ऐरे में से सबसे छोटा तत्व का चयन किया जाता है और इसे ऐरे में सबसे बाएँ तत्व

के साथ बदली किया जाता है, और वह तत्व सॉर्ट किये हुए ऐरे का हिस्सा बन जाता है। यह प्रक्रिया अवर्गीकृत ऐरे की सीमा को एक तत्व दायीं ओर बढ़ाती चली जाती है। इस एल्गोरिथ्म की औसत (average) और सबसे खराब (worst case) स्थिति में कॉम्प्लेक्सिटी $O(n)$ है, जहाँ n सॉर्ट किये जाने वाले तत्वों की संख्या है और इसलिए यह एल्गोरिथ्म बड़े डेटा सेट के लिए उपयुक्त नहीं है।

एक उदाहरण के रूप में निम्नलिखित दर्शाये हुए ऐरे पर विचार करते हैं :

14	33	27	10	35	19	42	44
----	----	----	----	----	----	----	----

क्रमबद्ध लिस्ट में प्रथम स्थान के लिए, पूरी लिस्ट की क्रमिक रूप से जांच होती है। पहली स्थिति जहाँ 14 को वर्तमान में संग्रहित करना है, हम पूरी लिस्ट को सर्च करते हैं और पाते हैं कि 10 निम्नतम मान है।

14	33	27	10	35	19	42	44
----	----	----	----	----	----	----	----

इसलिए हम 14 को 10 से बदलते हैं। एक पुनरावृत्ति के बाद 10 जो कि लिस्ट में न्यूनतम मान है, सॉर्टेड लिस्ट में पहली स्थिति में दिखाई देता है।

14	33	27	10	35	19	42	44
----	----	----	----	----	----	----	----

दूसरे स्थान के लिए जहाँ 33 है, हम एक रेखीय ढंग से बाकी लिस्ट की स्कैनिंग शुरू करते हैं।

10	33	27	14	35	19	42	44
----	----	----	----	----	----	----	----

हम पाते हैं कि 14 लिस्ट में दूसरा सबसे कम मान है और दूसरे स्थान पर होना चाहिए। हम इन मानों को स्वेप करते हैं।

10	33	27	14	35	19	42	44
----	----	----	----	----	----	----	----

दो पुनरावृत्तियों के बाद, दो कम से कम मान एक क्रमबद्ध ढंग से शुरुआत में आ जाते हैं।

10	14	27	33	35	19	42	44
----	----	----	----	----	----	----	----

यही प्रक्रिया ऐरे में बाकी के आइटम के लिए लागू की जाती है। पूरी सॉर्टिंग प्रक्रिया का एक सचित्र चित्रण निम्नानुसार है:

10	14	27	33	35	19	42	44
10	14	27	33	35	19	42	44
10	14	19	33	35	27	42	44
10	14	19	33	35	27	42	44
10	14	19	27	35	33	42	44
10	14	19	27	35	33	42	44
10	14	19	27	35	33	42	44
10	14	19	27	33	35	42	44
10	14	19	27	33	35	42	44

निवेशन (Insertion) सॉर्टिंग: यह एक इन-प्लेस तुलना-आधारित सॉर्टिंग एल्गोरिथ्म है। इसमें एक उप-लिस्ट बनाये रखी जाती है जो हमेशा सॉर्टेड रहती है। उदाहरण के लिए, एक ऐरे के निचले हिस्से को सॉर्टेड बनाए रखा जाता है। एक तत्व जिसे इस सॉर्टेड उप-लिस्ट में सम्मिलित किया जाना है, इसकी उचित जगह सर्च करके फिर इसे वहाँ डाला जाता है। इसलिए इसका नाम, निवेशन सॉर्ट है।

ऐरे को क्रमिक रूप से सर्च किया जाता है और अवर्गीकृत आइटम को स्थानांतरित किया जाता है उन्हें सॉर्टेड उप-लिस्ट में डाला (एक ही ऐरे में) जाता है। इस एल्गोरिथ्म की औसत (average) और सबसे खराब (worst case) स्थिति में कॉम्प्लेक्सिटी $O(n^2)$ है, जहाँ n सॉर्ट किये जाने वाले तत्वों की संख्या है और इसलिए यह एल्गोरिथ्म बड़े डेटा सेट के लिए उपयुक्त नहीं है। हम उदाहरण के लिए एक अवर्गीकृत ऐरे ले रहे हैं।

14	33	27	10	35	19	42	44
----	----	----	----	----	----	----	----

निवेशन सॉर्टिंग पहले दो तत्वों की तुलना करती है।

14	33	27	10	35	19	42	44
----	----	----	----	----	----	----	----

यहाँ 14 और 33 दोनों ही आरोही क्रम में पहले से ही हैं। अभी के लिए, 14 सॉर्टेड उप-लिस्ट में है।

14	33	27	10	35	19	42	44
----	----	----	----	----	----	----	----

निवेशन सॉर्टिंग आगे 33 की 27 से तुलना करती है।

14	33	27	10	35	19	42	44
----	----	----	----	----	----	----	----

और पाती है कि 33 सही स्थिति में नहीं है।

14	33	27	10	35	19	42	44
----	----	----	----	----	----	----	----

यह 33 को 27 के साथ स्वैप करती है।

यह सॉर्टेड उप-लिस्ट के सभी तत्वों के साथ की जाँच करती है। यहाँ हम देखते हैं कि सॉर्टेड उप-लिस्ट में केवल एक तत्व 14 है और 27, 14 से अधिक है, इसलिए सॉर्टेड उप-लिस्ट की अदला-बदली के बाद भी यह सॉर्टेड रहेगी।

14	27	33	10	35	19	42	44
----	----	----	----	----	----	----	----

अब तक सॉर्टेड उप-लिस्ट 14 और 27 है। इसके बाद, यह 33 की 10 से तुलना करती है।

14	27	33	10	35	19	42	44
----	----	----	----	----	----	----	----

यह मान एक सॉर्टेड क्रम में नहीं है।

14	27	33	10	35	19	42	44
----	----	----	----	----	----	----	----

इसलिए हम उन्हें स्वैप करते हैं।

14	27	10	33	35	19	42	44
----	----	----	----	----	----	----	----

हालांकि, स्वैपिंग 27 और 10 को अवर्गीकृत बनाता है।

14	27	10	33	35	19	42	44
----	----	----	----	----	----	----	----

इसलिए, हम उन्हें भी स्वैप करते हैं।

14	10	27	33	35	19	42	44
----	----	----	----	----	----	----	----

फिर हम 14 और 10 को अवर्गीकृत क्रम में पाते हैं।

14	10	27	33	35	19	42	44
----	----	----	----	----	----	----	----

हम उन्हें फिर से स्वेप करते हैं। तीसरी पुनरावृत्ति के अंत तक सॉर्टेड उप लिस्ट में 4 मान हो जाते हैं।

10	14	27	33	35	19	42	44
----	----	----	----	----	----	----	----

इस प्रक्रिया तब तक जारी रहती है जब तक सभी अवर्गीकृत मान सॉर्टेड उप-लिस्ट में शामिल नहीं हो जाते हैं।

प्रश्न 5. स्थिर और अस्थिर सॉर्टिंग के बीच क्या अन्तर है?

उत्तर- स्थिर (Stable) सॉर्टिंग और अस्थिर (Unstable) सॉर्टिंग

स्थिर सॉर्टिंग (Stable Sorting) : सॉर्टिंग एल्गोरिथ्म, तत्वों को सॉर्ट करने के बाद, एक जैसे तत्वों के क्रम जिसमें वो प्रकट होते हैं को परिवर्तित नहीं करती है उनको स्टेबल सॉर्टिंग कहा जाता है।

0	1	2	3	4	5	6	7	8	9
35	33	42	10	14	19	26	44	26	31

0	1	2	3	4	5	6	7	8	9
10	14	19	26	26	31	33	35	42	44

अस्थिर सॉर्टिंग (Unstable Sorting)

सॉर्टिंग एल्गोरिथ्म, तत्वों को सॉर्ट करने के बाद, एक जैसे तत्वों के क्रम जिसमें वो प्रकट होते हैं को परिवर्तित करती है। उनको अनस्टेबल सॉर्टिंग कहा जाता है।

0	1	2	3	4	5	6	7	8	9
35	33	42	10	14	19	26	44	26	31

0	1	2	3	4	5	6	7	8	9
10	14	19	26	26	31	33	35	42	44

एक एल्गोरिथ्म की स्टेबिलिटी (Stability) मायने रखती है जब हम मूल तत्वों का क्रम बनाए रखना चाहते हैं। उदाहरण के लिए एक टपल में।

अन्य महत्वपूर्ण प्रश्न

अतिलघु उत्तरीय प्रश्न

प्रश्न 1. सॉर्टिंग का क्या महत्त्व है?

उत्तर- सॉर्टिंग का सर्वाधिक महत्त्व डाटा सर्च को आसान बनाने में है।

प्रश्न 2. क्या इन-प्लेस सॉर्टिंग एल्गोरिथ्म को अतिरिक्त जगह की आवश्यकता होती है?

उत्तर- नहीं, इन-प्लेस सॉर्टिंग एल्गोरिथ्म को किसी भी अतिरिक्त जगह की आवश्यकता नहीं होती है।

प्रश्न 3. इन-प्लेस सॉर्टिंग का एक उदाहरण दीजिए।

उत्तर- बबल सॉर्ट इन-प्लेस सॉर्टिंग का एक उदाहरण है।

प्रश्न 4. क्या बबल सॉर्ट डेटा सेट के लिए उपयुक्त है?

उत्तर- नहीं बबल सॉर्ट बड़े डेटा सेट के लिए उपयुक्त नहीं है।

प्रश्न 5. बबल सॉर्ट की औसत (average) और सबसे खराब स्थिति में कॉम्प्लेक्सिटी क्या है?

उत्तर- बबल सॉर्ट की औसत (average) और सबसे खराब स्थिति में कॉम्प्लेक्सिटी $O(n^2)$ है। जहाँ n सॉर्ट किये जाने वाले तत्वों की संख्या है।

प्रश्न 6. चयन (Selection) सॉर्टिंग क्या है?

उत्तर- चयन सॉर्टिंग एक सरल सॉर्टिंग एल्गोरिथ्म है। यह एक इन-प्लेस तुलना-आधारित सॉर्टिंग एल्गोरिथ्म है।

प्रश्न 7. चयन (Selection) सॉर्टिंग की औसत (average) और सबसे खराब (worst case) स्थिति में कॉम्प्लेक्सिटी क्या है?

उत्तर- इस एल्गोरिथ्म की औसत (average) और सबसे खराब (worst case) स्थिति में कॉम्प्लेक्सिटी $O(n^2)$ है। जहाँ n सॉर्ट किये जाने वाले तत्वों की संख्या है।

प्रश्न 8. निवेशन (Insertion) सॉर्टिंग क्या है?

उत्तर- यह एक इन-प्लेस तुलना आधारित सॉर्टिंग एल्गोरिथ्म है। इसमें एक उप-लिस्ट बनाये रखी जाती है जो हमेशा सॉर्टेड रहती है।

प्रश्न 9. निवेशन (Insertion) सॉर्टिंग की औसत और सबसे खराब (worst case) स्थिति में कॉम्प्लेक्सिटी बताइए।

उत्तर- निवेशन (Insertion) सॉर्टिंग की औसत (average) और सबसे खराब (worst case) स्थिति में कॉम्प्लेक्सिटी $O(n^2)$ है। जहाँ n सॉर्ट किये जाने वाले तत्वों की संख्या है।

प्रश्न 10. त्वरित (Quick) सॉर्टिंग से आप क्या समझते हैं?

उत्तर- त्वरित (Quick) एक प्रकार की अत्यन्त कुशल सॉर्टिंग एल्गोरिथ्म है और डेटा के ऐसे को छोटे ऐसे में विभाजन करने पर आधारित है।

प्रश्न 11. त्वरित (Quick) सॉर्टिंग की औसत (average) और सबसे खराब (worst case) स्थिति में कॉम्प्लेक्सिटी बताइए।

उत्तर- त्वरित (Quick) सॉर्टिंग की औसत (average) और सबसे खराब (worst case) में कॉम्प्लेक्सिटी $O(n \log n)$ है।

लघु उत्तरीय प्रश्न

प्रश्न 1. अडैप्टिव और नॉन-अडैप्टिव सॉर्टिंग एल्गोरिथ्म के विषय में बताइए।

उत्तर- अडैप्टिव और नॉन-अडैप्टिव सॉर्टिंग (Adaptive and Non adaptive Sorting): यदि सॉर्टिंग एल्गोरिथ्म, सॉर्ट करने वाली लिस्ट में पहले से ही सॉर्टेड तत्वों का लाभ लेती है तब उसे अडैप्टिव कहा जाता है। अर्थात् सॉर्टिंग के दौरान यदि स्रोत (source) सूची में पहले से ही कुछ तत्व सॉर्टेड है तब अडैप्टिव एल्गोरिथ्म इसे ध्यान में रखते हुए उनका क्रम पुनः नहीं बदलती।

एक नॉन-अडैप्टिव सॉर्टिंग एल्गोरिथ्म सूची में पहले से ही सॉर्टेड तत्वों की ध्यान में नहीं रखती। वे तत्व सॉर्टेड है या नहीं की पुष्टि करने के लिए हर एक तत्व के क्रम को बदलती हैं।

प्रश्न 2. सॉर्टिंग तकनीकों में प्रयोग होने वाली कुछ शब्दावली का संक्षिप्त परिचय दीजिए।

अथवा

निम्नलिखित पर संक्षिप्त टिप्पणी लिखिए

(i) बढ़ता क्रम

- (ii) घटता क्रम
- (iii) गैर-बढ़ता क्रम
- (iv) गैर-घटता क्रम

उत्तर- सॉर्टिंग तकनीको पर चर्चा के दौरान आमतौर पर कुछ शब्दावली का प्रयोग किया जाता है, यहाँ उनका एक संक्षिप्त परिचय है:

बढ़ता क्रम (Increasing Order) : मानों का एक अनुक्रम बढ़ते हुए क्रम में कहा जाता है यदि बाद का तत्त्व अपने पिछले वाले तत्त्व से अधिक है। उदाहरण के लिए, 1, 3, 4, 6, 8, 9 बढ़ते क्रम में है, क्योंकि यहाँ हर अगला तत्त्व अपने पिछले वाले तत्त्व से अधिक है।

घटता क्रम (Decreasing Order) : मानों का एक अनुक्रम घटते हुए क्रम में कहा जाता है यदि बाद का तत्त्व अपने पिछले वाले तत्त्व से कम है। उदाहरण के लिए 9, 8, 6, 4, 3, 1, घटते क्रम में हैं क्योंकि यहाँ हर अगला तत्त्व अपने पिछले तत्त्व से कम है।

गैर-बढ़ता क्रम (Non-increasing Order): मानों का एक अनुक्रम गैर-बढ़ते हुए क्रम में कहा जाता है, यदि बाद का तत्त्व अपने पिछले वाले तत्त्व से कम या उसके बराबर है। यह क्रम तब होता है अनुक्रम में डुप्लिकेट मान हो। उदाहरण के लिए 9, 8, 6, 3, 3, 1, गैर बढ़ते क्रम में हैं। क्योंकि यहाँ हर अगला तत्त्व अपने पिछले तत्त्व से कम या उसके बराबर (3 के मामले में) है।

गैर-घटता क्रम (Non-increasing Order) : मानों का एक अनुक्रम गैर-घटते हुए क्रम में कहा जाता है, यदि बाद का तत्त्व अपने पिछले वाले तत्त्व से अधिक या उसके बराबर है। यह क्रम तब होता है जब अनुक्रम में डुप्लिकेट मान हो। उदाहरण के लिए 1, 3, 3, 6, 8, 9 गैर-घटते क्रम में हैं क्योंकि हर अगला तत्त्व अपने पिछले तत्त्व से अधिक या उसके बराबर (3 के मामले में) है।

प्रश्न 3. बबल सॉर्ट के लिए एल्गोरिथ्म लिखिए।

उत्तर- Algorithm : हम यहाँ यह मान रहे हैं कि तत्वों की लिस्ट एक ऐरे में है और स्वेप फंक्शन ऐरे तत्वों को स्वेप करता है।

Bubble Sort

for all elements of list

if list [i] > list[i+1]

swap (list[i], list [i+1])

end if

end for

return list

end Bubble Sort

प्रश्न 4. चयन (Selection) सॉर्ट के लिए एल्गोरिथ्म लिखिए।

उत्तर- एल्गोरिथ्म :

Step 1 – Set MIN to location 0

Step 2 – Search the minimum element in the list

Step 3 – Swap with value at location MIN

Step 4 – Increment MIN to point to next element

Step 5 – Repeat until list is sorted

प्रश्न 5. निवेशन (Insertion) सॉर्ट के लिए एल्गोरिथ्म लिखिए।

उत्तर- एल्गोरिथ्म

चरण 1 – अगर लिस्ट में केवल एक ही तत्व है, तो यह लिस्ट सॉर्टेड है।

चरण 2 – अगला तत्व लें।

चरण 3 – सॉर्टेड उप सूची के उन सभी तत्वों को शिफ्ट करें जो सॉर्ट किये जाने वाले मान से अधिक है।

चरण 5 – मान सम्मिलित करें।

चरण 6 – दोहराएँ जब तक सूची सॉर्ट नहीं हो जाता है।

प्रश्न 6. त्वरित सॉर्ट के लिए एल्गोरिथ्म लिखिए।

उत्तर- त्वरित सॉर्टिंग एल्गोरिथ्म : त्वरित एल्गोरिथ्म का उपयोग बारी-बारी से करते हैं जब तक कि हम छोटे सम्भव विभाजन तक नहीं पहुँच जाते। फिर प्रत्येक विभाजन पर त्वरित सॉर्ट की कारवाई की जाती है। हम निम्न रूप में त्वरित सॉर्ट की एल्गोरिथ्म को परिभाषित करते हैं

चरण 1 – सबसे दाएँ सूचकांक मान को पाइवोट बनाएँ।

चरण 2 – पाइवोट मान का उपयोग कर एरे का विभाजन करें।

चरण 3 – रिक्रसीवेली बाएँ विभाजन पर त्वरित सॉर्ट लगाएँ।

चरण 4 – रिक्रसीवेली दाएँ विभाजन पर त्वरित सॉर्ट लगाएँ।

निबंधात्मक प्रश्न

प्रश्न 1. बबल (Bubble) सॉर्ट को विस्तार से समझाइए।

उत्तर- बबल (Bubble) सॉर्ट: बबल सॉर्ट एक साधारण सॉर्टिंग एल्गोरिथ्म है। यह सॉर्टिंग एल्गोरिथ्म, तुलना-आधारित एल्गोरिथ्म है जिसमें सन्निकट तत्वों के प्रत्येक जोड़े की तुलना की जाती है और अगर वे क्रम में नहीं हैं तब तत्वों को बदला जाता है। इस एल्गोरिथ्म की औसत (average) और सबसे खराब (worst case) स्थिति में कॉम्प्लेक्सिटी $O(n^2)$ है जहाँ n सॉर्ट किये जाने वाले तत्वों की संख्या है और इसलिए यह एल्गोरिथ्म बड़े डेटा सेट के लिए उपयुक्त नहीं है।

बबल सॉर्टिंग कैसे काम करती है?

हम उदाहरण के लिए एक अनसॉर्टेड ऐरे ले रहे हैं। बबल सॉर्ट $O(n^2)$ समय लेती है, इसलिए हम इसे छोटा और सटीक रख रहे हैं।

14	33	27	35	10
----	----	----	----	----

बबल सॉर्ट, सबसे पहले दो तत्वों के साथ शुरू होती है, कौन-सा बड़ा है यह जाँच करने के लिए उनकी तुलना करती हैं।

14	33	27	35	10
----	----	----	----	----

इस मामले में, 33 मान 14 से अधिक है, इसलिए यह पहले से सॉर्टेड है। आगे हम 27 से 33 की तुलना करते हैं।

14	33	27	35	10
----	----	----	----	----

हम पाते हैं कि 27, 33 से छोटा है और इन दोनों मानों को बदला जाना चाहिए।

14	33	27	35	10
----	----	----	----	----

नई ऐरे इस तरह दिखानी चाहिए

14	27	33	35	10
----	----	----	----	----

आगे हम 33 और 35 की तुलना में पाते हैं कि दोनों पहले से ही सॉर्टेड स्थितियों में हैं।

14	27	33	35	10
----	----	----	----	----

आगे हम अगले दो मानों, 35 और 10 को देखते हैं।

14	27	33	35	10
----	----	----	----	----

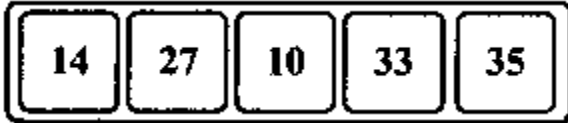
हम जानते हैं कि, 10, 35 से छोटा है इसलिए वे सॉर्टेड नहीं है।

14	27	33	35	10
----	----	----	----	----

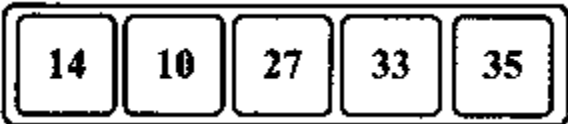
हम इन मानों को स्वेप करते हैं। हम पाते हैं कि हम ऐरे के अंत तक पहुँच चुके हैं। एक पुनरावृत्ति (iteration) के बाद, ऐरे इस तरह दिखना चाहिए



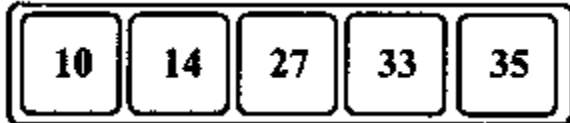
अब हम दिखा रहे हैं कि एक ऐरे प्रत्येक पुनरावृत्ति के बाद किस तरह दिखना चाहिए। दूसरी पुनरावृत्ति के बाद यह इस तरह दिखना चाहिए



ध्यान दें कि प्रत्येक पुनरावृत्ति के बाद, ऐरे के अंत में कम से कम एक मान चलता जाता है।



और जब किसी स्वेप की आवश्यकता नहीं रहती तब बबल सॉर्ट यह जान जाता है कि ऐरे पूरी तरह से सॉर्ट हो गया है।



प्रश्न 2. बबल सॉर्ट के द्वारा ऐरे के तत्वों को सॉर्ट करने का C प्रोग्राम लिखिए।

उत्तर- बबल सॉर्ट के लिए C प्रोग्राम:

```
#include<stdio.h>
#include<stdbool.h>
#define MAX 10
int list [MAX] = {1, 8, 4, 6, 0, 3, 5, 2, 7, 9};
void display ()
{
    int i;
    printf("[");
    // navigate through all items
    for ( i = 0; i < MAX; i ++ )
    {
        printf("%d",list[i]);
        printf("]\n");
    }
}
void bubble Sort ()
{

```



```

int temp;
int i, j;
bool swapped = false;
// loop through all numbers
for (i = 0; i < MAX-1; i++)
{
    swapped = false;
    // loop through numbers falling ahead
    for (j = 0; j < MAX-1-i; j++)
    {
        printf("Items compared: [%d, %d]", list[j], list[j+1]);
        // check if next number is lesser than current number
        // swap the numbers.
        // (Bubble up the highest number)
        if (list[j] > list[j+1])
        {
            temp=list[j];
            list[j]=list[j+1];
            list[j+1]=temp;
            swapped=true;
            printf("=> swapped [%d,%d] \n", list[j], list[j+1]);
        }
        else
        {
            printf ("=>not swapped\n");
        }
    }
    //if no number was swapped that means
    // array is sorted now, break the loop.
    if(!swapped)
    {
        break;
    }
    printf("Iteration %d#:", (i+1));
    display();
}
}
main()
{
    printf("Input Array:");

```

```
display();  
printf("\n");  
bubbleSort ();  
printf ("\nOutput Array:");  
display();  
}
```

जब उपरोक्त कोड कम्पायल और रन होगा तो आउटपुट निम्नानुसार होगा:

Input Array: [1 8 4 6 0 3 5 2 7 9]

Items compared: [1,8]=> not swapped

Items compared: [8,4]=> swapped [4,8]

Items compared: [8, 6]=> swapped [6,8]

Items compared: [8, 0]=> swapped [0, 8]

Items compared: [8, 3]=> swapped [3, 8]

Items compared: [8, 5]=> swapped [5,8]

Items compared: [8, 2]=> swapped [2, 8]

Items compared: [8, 7]=> swapped [7,8]

Items compared: [8, 9]=> not swapped

Iteration 1 #: [1460352789]

Items compared: [1,4]=> not swapped

Items compared: [4, 6] => not swapped

Items compared: [6,0]=> swapped [0,6]

Items compared: [6, 3]=> swapped [3, 6]

Items compared: [6,5]=> swapped [5,6]

Items compared: [6, 2]=> swapped [2, 6]

Items compared: [6, 7] => not swapped

Items compared: [7,8]=> not swapped

Iteration 2#: [1403526789]

Items compared: [1,4]=> not swapped

Items compared: [4,0]=> swapped [0,4]

Items compared: [4, 3]=> swapped [3, 4]

Items compared: [4, 5]=> not swapped

Items compared: [5,2]=> swapped [2,5]

Items compared: [5, 6]=> not swapped

Items compared: [6, 7]=> not swapped

Iteration 3#: [10342 56789]

Items compared: [1,0]=> swapped [0, 1]

Items compared: [1,3] => not swapped
Items compared: [3, 4] => not swapped
Items compared: [4,2]=> swapped [2,4]
Items compared: [4, 5]=> not swapped
Items compared: [5,6]=> not swapped

Iteration 4#: [01 32456789]

Items compared: [0, 1]=> not swapped
Items compared: [1,3]=> not swapped
Items compared:[3, 2]=> swapped [2,3]
Items compared: [3,4]=> not swapped
Items compared: [4, 5]=> not swapped

Iteration 5#: [0123456789]

Items compared: [0, 1]=> not swapped
Items compared: [1, 2]=> not swapped
Items compared: [2, 3]=> not swapped
Items compared: [3, 4]=> not swapped

Output Array: [0 1 2 3 4 5 6 7 8 9]

प्रश्न 3. चयन सॉर्ट द्वारा ऐरे के तत्वों को सॉर्ट करने का C प्रोग्राम लिखिए।

उत्तर- चयन सॉर्ट के लिए C प्रोग्राम:

```
#include<stdio.h>
#include<stdbool.h>
#define MAX 7
int int Array [MAX] = {4, 6, 3, 2, 1, 9, 7};
void printline (int count)
{
    int i;
    for(i = 0; i<count-1;i++)
    {
        printf("=");
    }
    printf("\n");
}
void display()
```

```

{
int i;
printf("[");
// navigate through all items
for(i=0; i<MAX; i++)
{
printf("%d", intArray[i];
}
printf("]\n");
}
void selection Sort()
{
int indexMin, i, j;
//loop through all numbers
for (i=0;i<MAX-1; i++)
{
// set current element as minimum
indexMin =i;
// check the element to be minimum
for(j=i+1;j<MAX; j++)
{
if(int Array [j] < int Array (index Min))
{
indexMin=j;
}
}
if(indexMin !=i)
{
printf("Items swapped: [%d,%d] \n", intArray[i], intArray[index Min]);
// swap the numbers
int temp = int Array[index Min];
int Array[index Min] = int. Array[i];
int Array[i] = temp;
}
printf("Iteration %d#:", (i+1));
display();
}
}
main()
{

```

```
printf("Input Array:");
display();
printfline (50);
selection Sort();
printf("Output Array:");
display();
printfline (50);
}
```

जब उपरोक्त कोड कम्पायल और रन होगा तो आउटपुट निम्नानुसार होगा:

Output

Input Array: [4632197]

=====

Items swapped: [4,1]

Iteration 1#:[1632497]

Items swapped: [6,2]

Iteration 2#:[1 236497]

Iteration 3#: [1 236497]

Items swapped: [6,4]

Iteration 4#:[1 234697]

Iteration 5#: [1 234697]

Items swapped: [9,7]

Iteration 6#:[1 2 34679]

Output Array: [1 2 3 46 79]

प्रश्न 4. मर्ज सॉर्ट द्वारा ऐरे के तत्वों को सॉर्ट करने का C प्रोग्राम लिखिए।

उत्तर- मर्ज सॉर्ट के लिए C प्रोग्राम:

```
#include<stdio.h>
#define max 10
int a[10] = {10, 14, 19, 26, 27, 31, 33, 35, 42, 44};
int b[10];
void merging(int low, int mid, int high)
{
    int i1, i2, i;
    for (i1=low, i2 = mid + 1, i = low; i1 <= mid && i2 <= high; i++)
    {
        if(a[i1] <= a[i2])
            b[i] = a[i1++];
    }
}
```

```

else
b[i] = a[12++];
}
while (11 <= mid)
b[i++] = a[11++];
while (12 <= high)
b[i++] = a[12++];
for (i = low; i <= high; i++)
a[i] = b[i];
}
void sort (int low, int high)
{
int mid;
if (low < high)
{
mid = (low + high)/2;
sort (low, mid);
sort (mid + 1, high);
merging (low, mid, high);
}
else
{
return;
}
}
int main()
{
int i;
printf("List before sorting \n");
for (i = 0; i <= max; i++)
printf("%d", a[i]);
sort (0, max);
printf("\nList after sorting\n");
for(i= 0; i <=max; i++)
printf("%d", a[i]);
}

```

जब उपरोक्त कोड कम्पायल और रन होगा तो आउटपुट निम्नानुसार होगा:

List before sorting

10 14 19 26 27 31 33 35 42 44 0

List after sorting

0 10 14 19 26 27 31 33 35 42 44

प्रश्न 5. निवेशन (Insertion) सॉर्ट द्वारा ऐरे के तत्वों को सॉर्ट करने का C प्रोग्राम लिखिए।

उत्तर- निवेशन (Insertion) सॉर्टिंग के लिए C प्रोग्राम:

```
#include <stdio.h>
#include
#define MAX 7
int int Array [MAX] = {4, 6, 3, 2 1, 9, 7};
void printline (int count)
{
    int i;
    for(i=0;i<count-1; i++)
    {
        printf("=");
    }
    printf("\n");
}
void display()
{
    int i;
    printf("[");
    // navigate through all items
    for(i=0; i<MAX; i++)
    {
        printf("%d", int Array[i]);
    }
    printf("]\n");
}
void insertion Sort()
{
    int value To Insert;
    int hole Position;
    int i;
    // loop through all numbers
    for(i=1; i < MAX; i++)
    {
```

```

// select a value to be inserted.
value To Insert = int Array [i];
// select the hole position where number is to be inserted
hole Position = i;
// check if previous no. is larger than value to be inserted
while (hole Position > 0 && int Array[hole Position-1] > value To Insert)
{
int Array [hole Position] = int Array [hole Position-1];
hole Position- -;
printf("item moved: %d\n", int Array[hole Position]);
}
if (hole Position != i)
{
printf("item inserted : %d, at position : %d\n", value To Insert, hole Position);
// insert the number at hole position
int Array [hole Position] = value To Insert;
}
printf ("Iteration %d#:", i);
display();
}
}
main()
{
printf("Input Array:");
display();
printline (50);
insertion Sort();
printf("Output Array:");
display();
printline (50);
}

```

जब उपरोक्त कोड कम्पायल और रन होगा तो आउटपुट निम्नानुसार होगा:

Input Array: [4632197]

= = = = =

Iteration 1#: [4632197]

item moved: 6

item moved: 4

item inserted : 3, at position : 0

Iteration 2#: [3462197]
item moved : 6
item moved :4
item moved :3
item inserted : 2, at position : 0

Iteration 2# : [2346197]
item moved : 6
item moved : 4
item moved :3
item moved : 2
item inserted : 1, at position : 0

Iteration 4#: [1 23469 7]
Iteration 5# : [1234697]
item moved : 9
item inserted : 7, at position : 5
Iteration 6#: [1 2 3 4679]
Output Array: [1 234679]

प्रश्न 6. त्वरित (Quick) सॉर्ट द्वारा ऐरे के तत्वों को सॉर्ट करने का C प्रोग्राम लिखिए।

उत्तर- त्वरित सॉर्टिंग के लिए C प्रोग्राम:

```
#include
#include
#define MAX 7
int intArray[MAX] = {4, 6, 3, 2, 1, 9, 7};
void printline (intcount)
{
    int i;
    for(i = 0; i < count-1;i++)
    {
        printf("=");
    }
    printf ("=\n");
}
void display()
{
```

```

int i;
printf("[");
// navigate through all items
for(i = 0; i<MAX; i++)
{
printf("%d", intArray[i];
}
printf("]\n");
}
void swap (int num1, int num2)
{
int temp = intArray[num1];
intArray[num1] = intArray[num2];
intArray[num2] = temp;
}
int partition (int left, int right, int pivot)
{
int leftPointer = left-1;
int rightPointer = right;
while (true)
{
while (int Array[++leftPointer] < pivot)
{
//do nothing
}
while (right Pointer > 0 && intArray[--rightPointer] > pivot)
{
// do nothing
}
if(left Pointer > = right Pointer)
{
break;
}
else
{
printf("item swapped : %d, %d\n",)
intArray(left Pointer), intArray (rightPointer));
swap(leftPointer);
printf ("Updated Array:");
display();
}
}

```

```

return left Pointer;
}
void quick Sort (int left, int right)
{
if (right-left < = 0)
{
return;
}
else
{
int pivot = int Array(right);
int partition Point = partition (left, right, pivot);
quick Sort (left, partition Point-1);
}
}
main()
{
printf("Input Array:");
display();
printline (50);
quick sort (0, MAX-1);
printf("Output Array;");
display();
printline (50);
}

```

जब उपरोक्त कोड कम्पायल और रन होगा तो आउटपुट निम्नानुसार होगा:

```

Input Array: [4632197]
=====
pivot swapped: 9,7
Updated Array:[4632179]
pivot swapped : 4,1
Updated Arra: [1632479]
item swapped: 6,2
pivot swapped : 6,4
Updated Array: [1234679]
pivot swapped: 3,3
Updated Array: [1234679]
Output Array: [1234679]

```