

Understanding Sorting

सीबीएसई पाठ्यक्रम पर आधारित
कक्षा -11



अध्याय - 10



द्वारा:

संजीव भदौरिया

स्नातकोत्तर शिक्षक (संगणक विज्ञान)

के० वि० बाराबंकी (लखनऊ संभाग)

संजीव भदौरिया, के० वि० बाराबंकी

परिचय

- सामान्यतया sorting का अर्थ होता है, objects के प्रकार(type) के अनुसार objects को अलग अलग करके उनका समूह बनाना ।
- प्रोग्रामिंग में और कंप्यूटर में sorting का पृथक अर्थ होता है, जैसा कि हम जानते हैं कि कंप्यूटर को प्रायः विभिन्न प्रकार के elements के समूह के साथ कार्य करना होता है - और “ कंप्यूटर में sorting का अर्थ होता है उन elements को एक क्रम में व्यवस्थित करना ।”
- क्रम कोई भी हो सकता है घटते क्रम में या बढ़ते क्रम में ।
- इस अध्याय में हम sorting के बारे में जानेंगे और इसके लिए दो तकनीकों का प्रयोग करेंगे –
 1. Bubble Sort
 2. Insertion Sort

Sorting क्या है ?

- “ कंप्यूटर में sorting का अर्थ होता है उन elements को एक क्रम में व्यवस्थित करना |” क्रम कोई भी हो सकता है घटते क्रम में या बढ़ते क्रम में |
- माना एक sequence निम्न है -
Seq = [6,8,9,1,3,4]
तब इसका बढ़ते हुए क्रम में sort किया हुआ नया sequence निम्न होगा –
Seq = [1,3,4,6,8,9]
और घटते हुए क्रम में इसका sort किया हुआ नया sequence होगा –
Seq = [9,8,6,4,3,1]
- ऐसा करने के लिए कई तकनीकें हैं जैसे - *Selection Sort, Bubble Sort, Insertion Sort, Heap Sort, Quick Sort* इत्यादि
- परन्तु हमारे पाठ्यक्रम में सिर्फ दो प्रकार की तकनीकों का प्रयोग है -
- **Bubble sort**
- **Insertion sort**

Bubble Sort

- Bubble Sort में अगल बगल वाले दो values की आपस में तुलना करते हैं और यदि वे सही क्रम में न हों तो उनकी अदला बदली (Exchange) कर देते हैं | प्रत्येक पास में सबसे बड़ा अपनी निश्चित स्थिति की ओर बढ़ता रहता है |

```
lst=[10,4,6,18,2,17,3]
l=len(lst)
for i in range(l):
    for j in range(0,l-i-1):
        if lst[j]>lst[j+1]:
            temp=lst[j]
            lst[j]=lst[j+1]
            lst[j+1]=temp
print("After sorting the list is : ")
print(lst)
```

आप इस program को dry run करके आसानी से समझ सकते हैं | पाइथन के debugger tool से भी हम इसको आसानी से समझ सकते हैं

OUTPUT:

```
After sorting the list is :
[2, 3, 4, 6, 10, 17, 18]
```

Bubble Sort में operation की संख्या ज्ञात करना

- Operation की संख्या एक बहुत महत्वपूर्ण पहलू है क्योंकि ज्यादा operation का मतलब CPU समय का ज्यादा प्रयोग होता है |
- दो अलग अलग तर्क(logic) के program एक ही परिणाम अलग अलग operation की संख्या में निकाल सकते हैं |
- हम लो एक statement को एक operation मान के चलते हैं जिसमे assignment, comparision, swaping शामिल है |

```
1. lst=[10,4,6,18,2,17,3]
2. print("Original List is : ",lst)
3. l=len(lst)
4. for i in range(l):
5.     for j in range(0,l-i-1):
6.         if lst[j]>lst[j+1]:
7.             temp=lst[j]
8.             lst[j]=lst[j+1]
9.             lst[j+1]=temp
10. print("After sorting the list is : ")
11. print(lst)
```

Python इसकी (Swap) जगह एक और statement देता है -

```
lst [ j ], lst [ j+1 ] = lst [ j + 1 ], lst [ j ]
```

Bubble Sort में operation की संख्या ज्ञात करना

इसके operation पता करने के लिए -

- लाइन 1, लाइन 2, लाइन 3 में Operations = 3
- लाइन 4 से 9 तक के operation पता करने हैं = ?
- लाइन 10 में operation = 1
- अब outer for loop के operation पता करते हैं
- उदाहरण में दिए गए 7 मानों के लिए outer loop 7 बार चलेगा -
 - लाइन 4 में operation 1
 - लाइन 5 से 9 इनर for loop जो कि उदाहरण के अनुसार $(n-i-1)$ अर्थात 6 बार चलेगा।
 - लेकिन यह पहली बार 6 बार, दूसरी बार 5 बार, तीसरी बार 4 बार चलेगा इसी प्रकार चलता रहेगा।
 - लाइन 5 में operation = 1
 - लाइन 6,7,8,9 में operation = 4
- अतः outer for loop के operations हैं $\left. \begin{array}{l} \text{लाइन 4 में operation 1} \\ \text{लाइन 5 से 9 इनर for loop के operations 5} \end{array} \right\} \text{कुल 5}$
- 1st iteration में $(i=0) = 1 \text{ op (line 4)} + 5 \text{ ops (Line 5-9)} \times 6 \text{ बार} = 31 \text{ ops}$
- 2nd iteration में $(i=1) = 1 \text{ op (line 4)} + 5 \text{ ops (Line 5-9)} \times 5 \text{ बार} = 26 \text{ ops}$
- 3rd iteration में $(i=2) = 1 \text{ op (line 4)} + 5 \text{ ops (Line 5-9)} \times 4 \text{ बार} = 21 \text{ ops}$
- 4th iteration में $(i=3) = 1 \text{ op (line 4)} + 5 \text{ ops (Line 5-9)} \times 3 \text{ बार} = 16 \text{ ops}$
- 5th iteration में $(i=4) = 1 \text{ op (line 4)} + 5 \text{ ops (Line 5-9)} \times 2 \text{ बार} = 11 \text{ ops}$
- 5th iteration में $(i=5) = 1 \text{ op (line 4)} + 5 \text{ ops (Line 5-9)} \times 1 \text{ बार} = 6 \text{ ops}$
- 7th iteration में $(i=6) = 1 \text{ op (line 4)} + 5 \text{ ops (Line 5-9)} \times 0 \text{ बार} = 1 \text{ ops}$
- कुल मिलकर outer loop के 112 operation होंगे। इस प्रकार कुल operation $3+112+1 = 116$

Bubble Sort में operation की संख्या ज्ञात करना

- यदि हम निम्न program के अनुसार चलें तो -
 1. lst=[10,4,6,18,2,17,3]
 2. print("Original List is : ",lst)
 3. l=len(lst)
 4. for i in range(l):
 5. for j in range(0,l-i-1):
 6. if lst[j]>lst[j+1]:
 7. Lst [j], lst [j + 1] = lst [j + 1], lst [j]
 8. print("After sorting the list is : ",lst)
- N के 7 मानों के लिए इसमें 74 operations होंगे |
- N के 8 मानों के लिए 96, N के 9 मानों के लिए 121, N के 6 मानों के लिए 55 operations होंगे |
- N के कुल मानों के लिए comparision की संख्या होगी $(N-1)+(N-2)+(N-3) + \dots + (2)+(1)$ अर्थात N^2 .
- N के कुल मानों के लिए Swapping की संख्या होगी $(N-1)+(N-2)+(N-3) + \dots + (2)+(1)$ अर्थात N^2 .
- Best case में जब sequence पूरी तरह से sort होगी तब कुल ops होंगे N^2 .
- Worst केस में जब हर element उलटे क्रम में होगा तो कुल ops होंगे N^2 (comparision)+ N^2 (Swappings) = $2 N^2$

Insertion Sort

- Insertion Sort में हम पहली दो values को सही क्रम में लगा लेते हैं।
- इसे हम अपना sub-sequence कहेंगे जो कि sorted रहेगा।
- बाकी बचे unsorted sequence से एक-एक value लेते जायेंगे और sorted वाले sub-sequence में सही स्थान पर insert करते जायेंगे।
- और यह क्रम tab तक चलेगा जबतक कि पूरी sequence sort नहीं हो जाती।

```
lst=[15,14,19,2,6,4,5,11,12,10,8,7]
print("Original List is :", lst)
for i in range(1,len(lst)):
    temp=lst[i]
    j=i-1
    while j>=0 and temp<lst[j]:
        lst[j+1]=lst[j]
        j=j-1
    else:
        lst[j+1]=temp
print("List after sort is :",lst)
```

OUTPUT:

```
Original List is : [15, 14, 19, 2, 6, 4, 5, 11, 12, 10, 8, 7]
List after sort is : [2, 4, 5, 6, 7, 8, 10, 11, 12, 14, 15, 19]
```

आप इस program को dry run करके आसानी से समझ सकते हैं। पाइथन के debugger tool से भी हम इसको आसानी से समझ सकते हैं।

Bubble Sort और Insertion Sort के अनुप्रयोग

- Bubble sort को सबसे साधारण और आसान sorting algorithm माना जाता है परन्तु यह सबसे निपुण algorithm नहीं माना जाता है ।
- बल्कि छोटे छोटे database के लिए भी sorting में insertion sort, bubble sort से बेहतर है ।
- लेकिन जब किसी डाटा के समूह को किसी ऐसी file जिसे tape में संग्रहीत किया गया हो तो उस file के डाटा को sort करने में bubble sort सबसे बेहतर है । क्योंकि tape में डाटा को sequential रूप से संग्रहीत किया जाता है ।
- Insertion sort का प्रयोग प्रायः छोटे छोटे database को sort करने के लिए किया जाता है। उस स्थिति में यह तेज़ी के कार्य करता है जब database लगभग sort हो। इसमें ज्यादा मेमोरी स्पेस की आवश्यकता नहीं होती है ।
- छोटे sequences के लिए bubble sort की अपेक्षा insertion sort में कम swapping होती है ।

प्रश्नोत्तरी

- Sorting क्या है और इसके कुछ तकनीकों के नाम लिखिए ।
- Bubble sort के पीछे का सिद्धांत लिखिए ।
- किस आधार पर आप यह तय करते हैं कि कौन सी algorithm अच्छी है ?
- Insertion sort के पीछे का सिद्धांत लिखिए ।
- Operation की संख्या के आधार पर insertion sort और bubble sort की तुलना करिए ।
- किस केस में insertion sort भी दक्ष नहीं है ?
- Insertion sort और bubble sort तकनीकों में मुख्य असमानता क्या है ?

धन्यवाद

और अधिक पाठ्य-सामग्री हेतु निम्न लिंक पर क्लिक करें -

www.pythontrends.wordpress.com

एक शुरुआत pythontrends

पाइथन सीखें और सिखाएं

मुख्य पृष्ठ/Home

संपर्क/Contact

लेख/Articles

छायाचित्र/Images

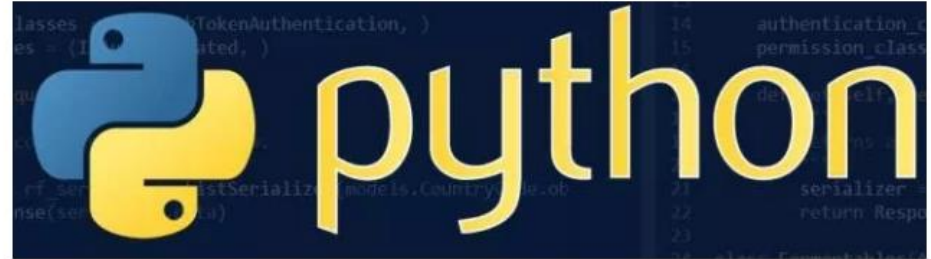
विडियो/Video

अध्यायवार पाठ्यसामग्री/Lesson wise
Study Material

उपयोगी लिंक्स / Useful Links

पाइथन प्रोग्राम/Python Programs

नमस्ते दोस्तों ! /Hello Friends!



यह ब्लॉग उन बच्चों की मदद के लिए बनाया गया है जो python में प्रोग्रामिंग सीख रहे हैं | यह ब्लॉग द्विभाषीय होगा जिससे सीबीएसई बोर्ड के वे बच्चे जिन्हें अंग्रेजी भाषा में समस्या होती है उन्हें सही मार्गदर्शन करेगा तथा प्रोग्रामिंग में उनकी सहायता करेगा | जैसा की हम जानते हैं की हमारे देश में कई क्षेत्र और कई लोग ऐसे हैं जिनकी अंग्रेज़ी उतनी मज़बूत नहीं है क्यों कि ये हमारी मातृभाषा नहीं है | तो हमें कभी कभी अंग्रेज़ी के कठिन शब्दों को

