Sorting Problem

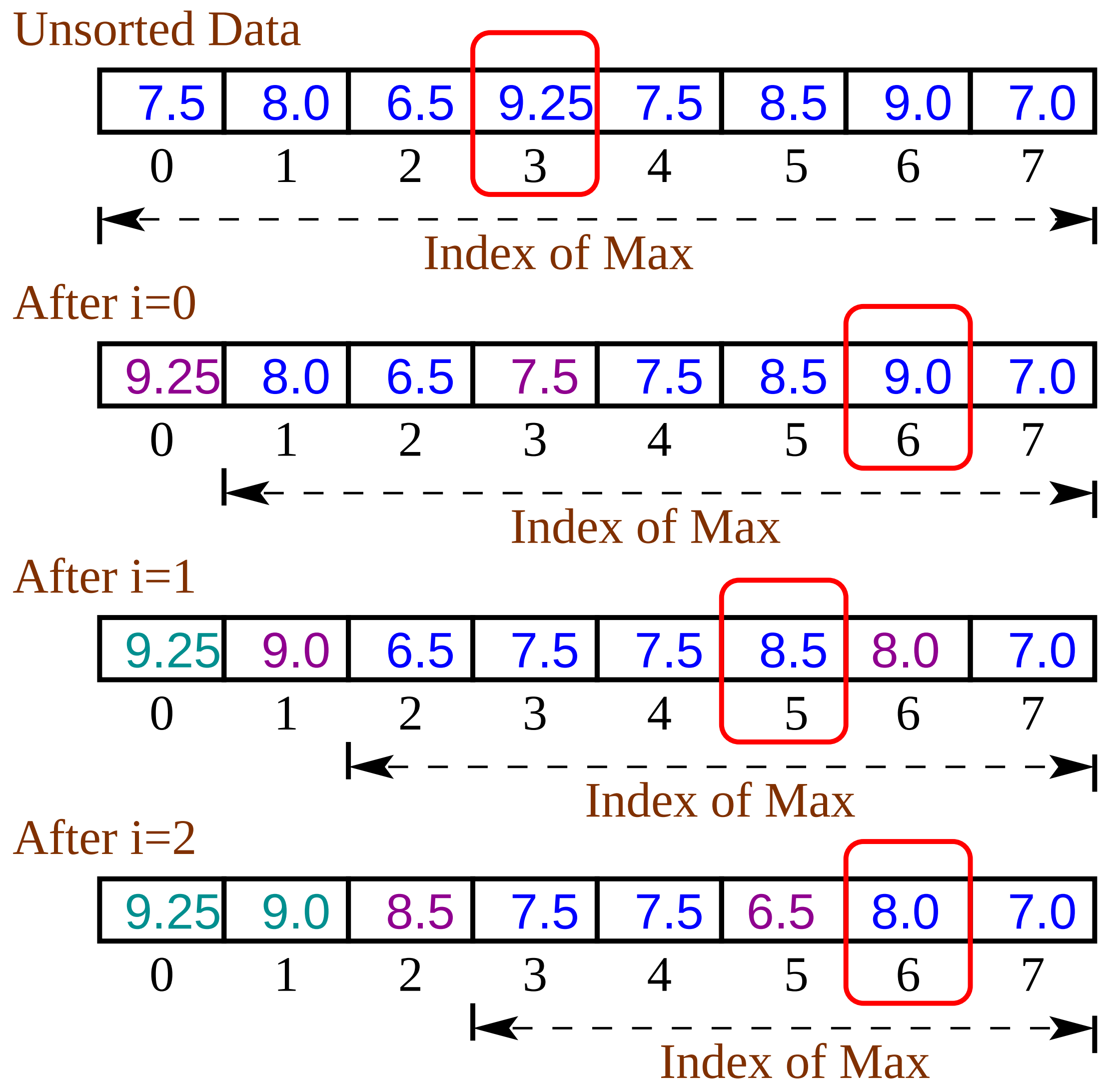
Unsorted Data									
7.5	8.0	8.5	8.25	9.25	9.0	6.5	8.0	7.0	7.5

Sorted Data (non-ascending: larger → smaller)									
9.25	9.0	8.5	8.25	8.0	8.0	7.5	7.5	7.0	6.5

Sorted Data (non-descending: smaller → larger)									
6.5	7.0	7.5	7.5	8.0	8.0	8.25	8.5	9.0	9.25

Selection Sort

The data is stored in a list and we sort them in non-ascending order. Let the length of the list be n.

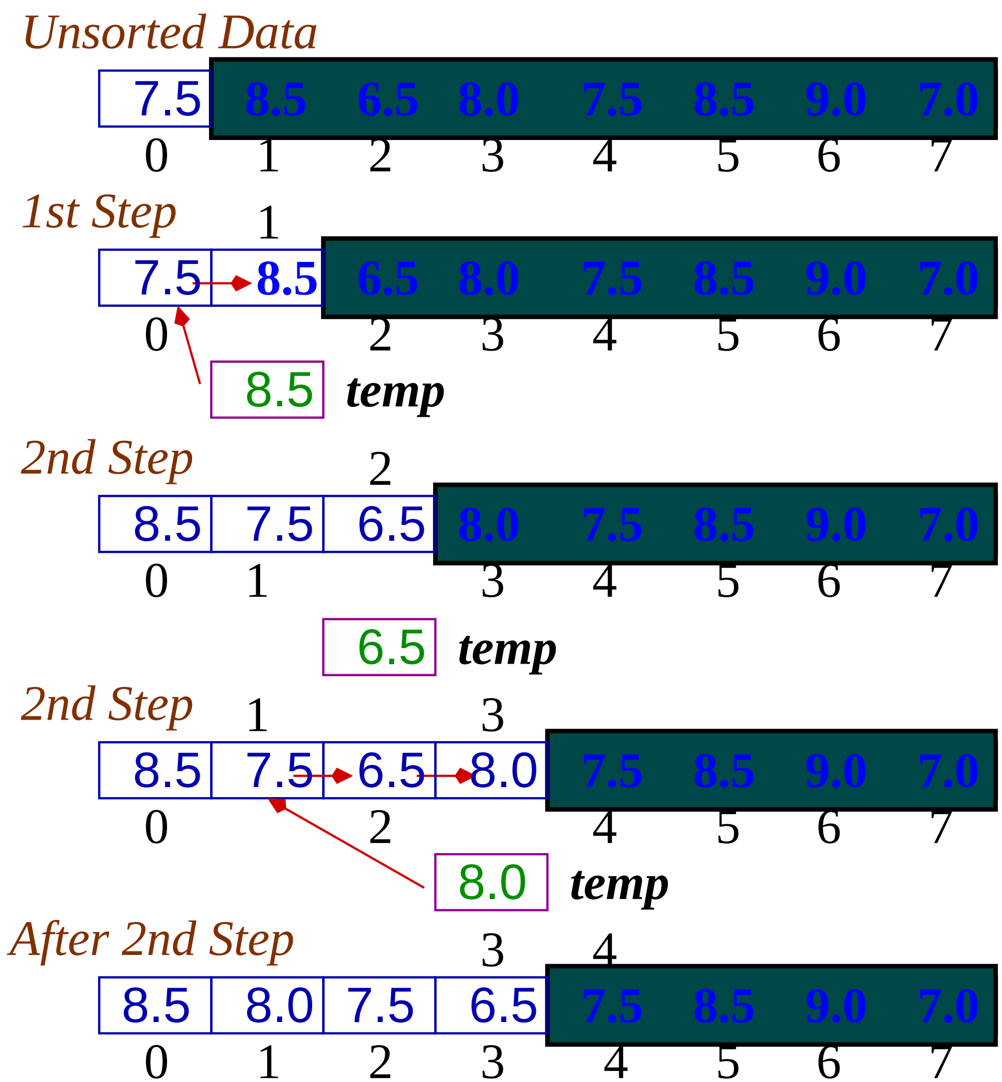

Unsorted Data
7.5 8.0 6.5 9.25 7.5 8.5 9.0 7.0
0 1 2 3 4 5 6 7
Index of Max
After i=0
9.25 8.0 6.5 7.5 7.5 8.5 9.0 7.0
0 1 2 3 4 5 6 7
Index of Max
After i=1
9.25 9.0 6.5 7.5 7.5 8.5 8.0 7.0
0 1 2 3 4 5 6 7
Index of Max
After i=2
9.25 9.0 8.5 7.5 7.5 6.5 8.0 7.0
0 1 2 3 4 5 6 7
Index of Max

Selection Sort Algorithm

```
for i ← 0 to n − 2 do
    max ← a[i], maxIndex ← i
    for j ← i + 1 to n − 1 do
        if max < a[j] then
            max ← a[j], maxIndex ← j
        endIf
    endFor
    a[i] ↔ a[maxIndex]      #Exchange
endFor
```

Insertion Sort

The problem is identical but the method is different.


Unsorted Data
7.5 8.5 6.5 8.0 7.5 8.5 9.0 7.0
0 1 2 3 4 5 6 7
1st Step
1
7.5 8.5 6.5 8.0 7.5 8.5 9.0 7.0
0 2 3 4 5 6 7
8.5 temp
2nd Step
2
8.5 7.5 6.5 8.0 7.5 8.5 9.0 7.0
0 1 3 4 5 6 7
6.5 temp
2nd Step
1 3
8.5 7.5 6.5 8.0 7.5 8.5 9.0 7.0
0 2 4 5 6 7
8.0 temp
After 2nd Step
3 4
8.5 8.0 7.5 6.5 7.5 8.5 9.0 7.0
0 1 2 3 4 5 6 7

Insertion Sort Algorithm

```
for i ← 1 to n −1 do
    temp ← l[i]
    for j ← i-1 downto 0 do
        if l[j] > temp
            l[j+1] ← l[j]
        else go out of loop
    endFor
    l[j+1] ← temp
endFor
```

Computation of $sin(x)$

Power Series

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} \cdots$$

Finite number of terms of this infinite series may be used to compute an approximate value of $\sin(x)$, where x is in radian.

π radian is 180°

$$
\begin{aligned}
\sin x &= x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} \cdots \\
&= \sum_{i \geq 0} (-1)^i \frac{x^{2i+1}}{(2i+1)!} \\
&= \sum_{i \geq 0} t_i, \text{ where } t_i = (-1)^i \frac{x^{2i+1}}{(2i+1)!}
\end{aligned}
$$

Inductive Definition of t_i

$$t_i = \begin{cases} x & \text{if } i = 0, \\ -t_{i-1}\dfrac{x^2}{2i(2i+1)} & \text{if } i > 0. \end{cases}$$

This is also called recurrence relation or recursive definition.

Approximation of $\sin(x)$

The sum up to the n^{th} term (S_n) of the series gives an approximate value of $\sin(x)$. The inductive definition of S_n is

$$S_n = \begin{cases} t_0 & \text{if } n = 0, \\ s_{n-1} + t_n & \text{if } n > 0. \end{cases}$$

From Inductive Definition to Iterative Process

An iterative process of computation can be obtained from the inductive definition.

1. Start from initial values t_0 and S_0.
2. Repeat the following two steps.
   (a) Compute the next term, $t_{i+1},\ i = 0, 1, \cdots$.
   (b) Compute the next approximate value of $\sin(x)$ by computing $S_{i+1},\ i = 0, 1, \cdots$.

Termination of Iteration

The process is to be terminated after a finite number of iterations. The termination may be

1. after a fixed number of iterations, or
2. after achieving a pre-specified accuracy.

Fixed No of Iterations

```
# sin1.py : computation of sin x
import math
def mySin(x):
    x = x%(2.0*math.pi)
    term = x
    termSum = term
    for i in range(100)[1:]:
        factor = 2.0*i
        factor = factor*(factor+1.0)
```

```
        factor = -x*x/factor
        term = term * factor
        termSum = termSum + term
    return termSum
a = input("Input angle in degree: ")
x = math.pi*a/180.0
print "sin(", x, ") = ", mySin(x)
```

Problem 8

Write a Python program that reads a positive integer n and prints the list of all its factors in pairs.

Input: 24

Output: (1, 24) (2, 12) (3, 8) (4, 6)

Input: 36

Output: (1, 36) (2, 18) (3, 12) (4, 9) 6

Problem 9

Write a Python program that reads a positive integer n. It prints all integers in the range $1 \cdots n$, that has exactly three factors.

Input: 50

Output: 4, 9, 25, 49

Problem 10

Write a Python program that reads a list of positive integers. It prints all pairs of data p, q from the list that are relatively prime i.e. $\gcd(p, q) = 1$. You may use the gcd function we supplied earlier.

Input: [24, 15, 35, 43, 18]

Output: (24, 35) (24, 43) (15, 43) (35, 43) (35, 18) (43, 18)

Inner Product

Let $p = [p_0, p_1, \cdots, p_{n-1}]$ and $q = [q_0, q_1, \cdots, q_{n-1}]$ be two lists of numbers of equal lengths. The inner product of p and q is $p \cdot q = p_0 \times q_0 + p_1 \times q_1 + \cdots + p_{n-1} \times q_{n-1}$.

Problem 11

Write a recursive Python function `innerProd(p,q)` that takes two lists of numbers and returns the inner product of p and q. Complete the program to test the function.
Input: [1,2,3] [2,3,4]
Output: 20

Problem 12

Write a recursive Python function `maxD(l)` that takes a non-null list of numbers and returns the largest among them. Complete the program to test the function.

Input: $[23, -11, 2, 72, -52]$

Output: 72

Problem 13

What does the following function do when called as `app(fact,l)` where `fact(n)` computes factorial and `l` is a list of positive integers.

```
def app(f,l):
    if l == []: return []
    return [f(l[0])] + app(f,l[1:])
```

Complete the program.

Problem 14

What does the following function do?

```
def app(l, x):
    if l == []: return []
    return [l[0](x)] + app(l[1:],x)
```

Rational as Continued Fraction

A rational number, $\frac{p}{q}$, where p, q are non-negative integers and $q \neq 0$, may be expressed as a continued fraction. Following is an example for $\frac{61}{27}$.

Rational as Continued Fraction

$$\frac{61}{27} = 2 + \cfrac{1}{3 + \cfrac{1}{1 + \cfrac{1}{6}}}$$

This may be represented in a compact form as a list $[2, 3, 1, 6]$, where the first 2 indicates the integral part of the fraction.

Problem 15

Write a Python program that reads the numerator and denominator of a rational number. Calls a function `ratToCF(num, den)`. that returns the simple continued fraction as a list. It is then printed.

Input: 23, 34

Output: [0, 1, 2, 11]

Problem 16

Implement selection sort in a Python program.
Input: 2 5 -1 7 23 16
Output: [23,16,7,5,2,-1]

Problem 17

Implement Insertion sort as a Python function `insSort(l)`. Read the data, prepare a list and pass it as a parameter to `insSort(l)`. Then print the sorted data.

Input: 2 5 -1 7 23 16

Output: [23,16,7,5,2,-1]

Computation of $cos(x)$

Power Series

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} \cdots$$

Finite number of terms of this infinite series may be used to compute an approximate value of $\cos(x)$, where x is in radian.

π radian is $180°$

Problem 18

Write the Python function `myCos(x)`. Use it to print the table of $\cos\theta$, $\theta = 0^\circ$ to 180° at an interval of 10°.