problem
solving
using
computers

CSE 1051

Selection Sorting
S13_2

Objectives

To learn and appreciate the following concepts

Sorting Technique

- Selection Sort

- Selection Sort with Strings

Session Outcome

- At the end of the session the student will be able to understand:

 - Importance of selection sorting on integers and strings

Selection Sort

- Each pass selects the smallest data item from the unsorted set and move it to its position

- **Procedure:**

  Here 'N' indicates the number of data items to be sorted

  - **Step 1:** From the data items in positions 0 to N-1, select the smallest data item and interchange with the 0th data item. Now the first data item is sorted

  - **Step 2:** From the data items in positions 1 to N-1, select the smallest data item and interchange with the 1st data item. Now the second data item is sorted

  - **Step 3:** The steps are repeated N-1 times. At the end of N-1 th time the entire data set is sorted

Selection Sort – example

Pass 1

Selection Sort – example

Pass 2
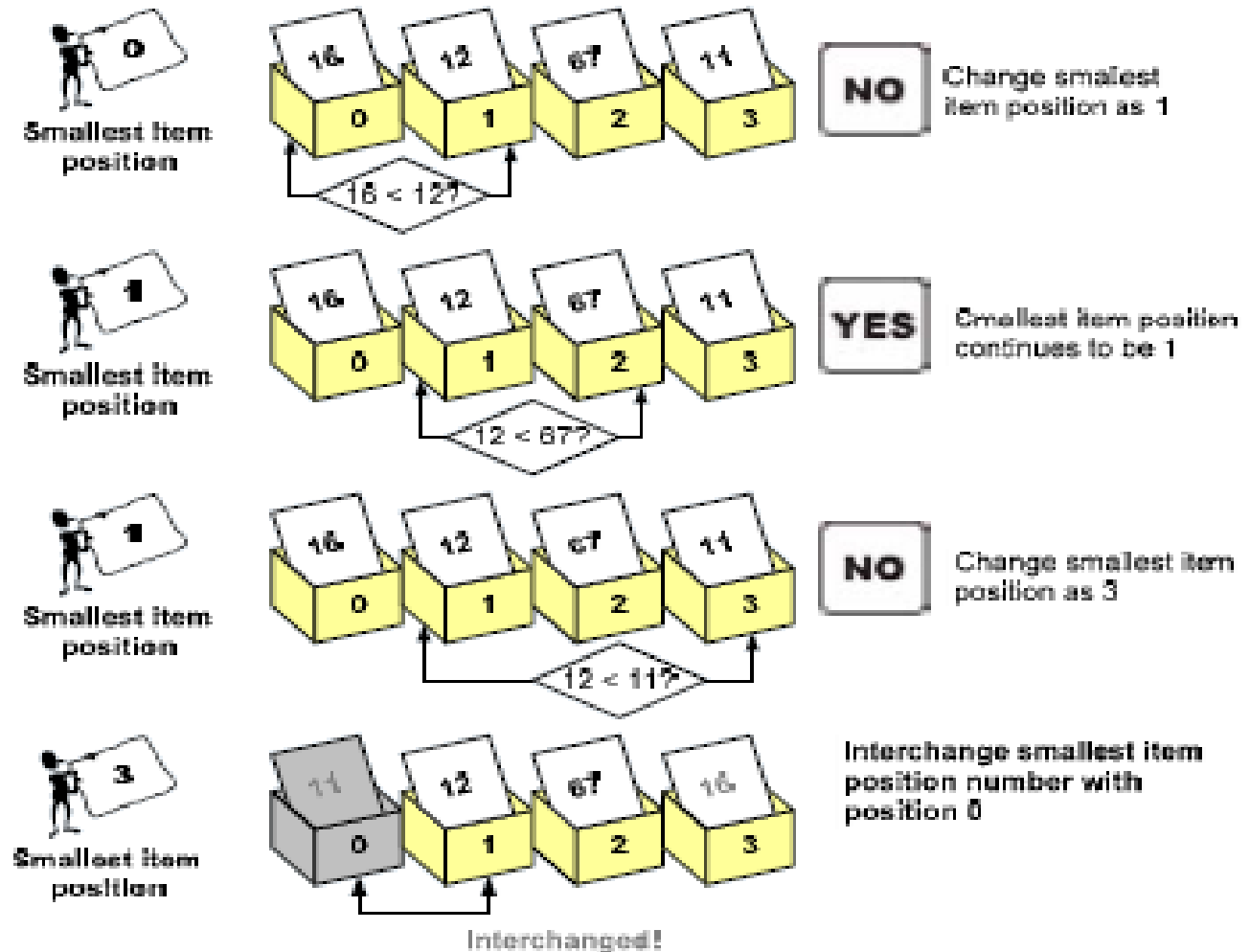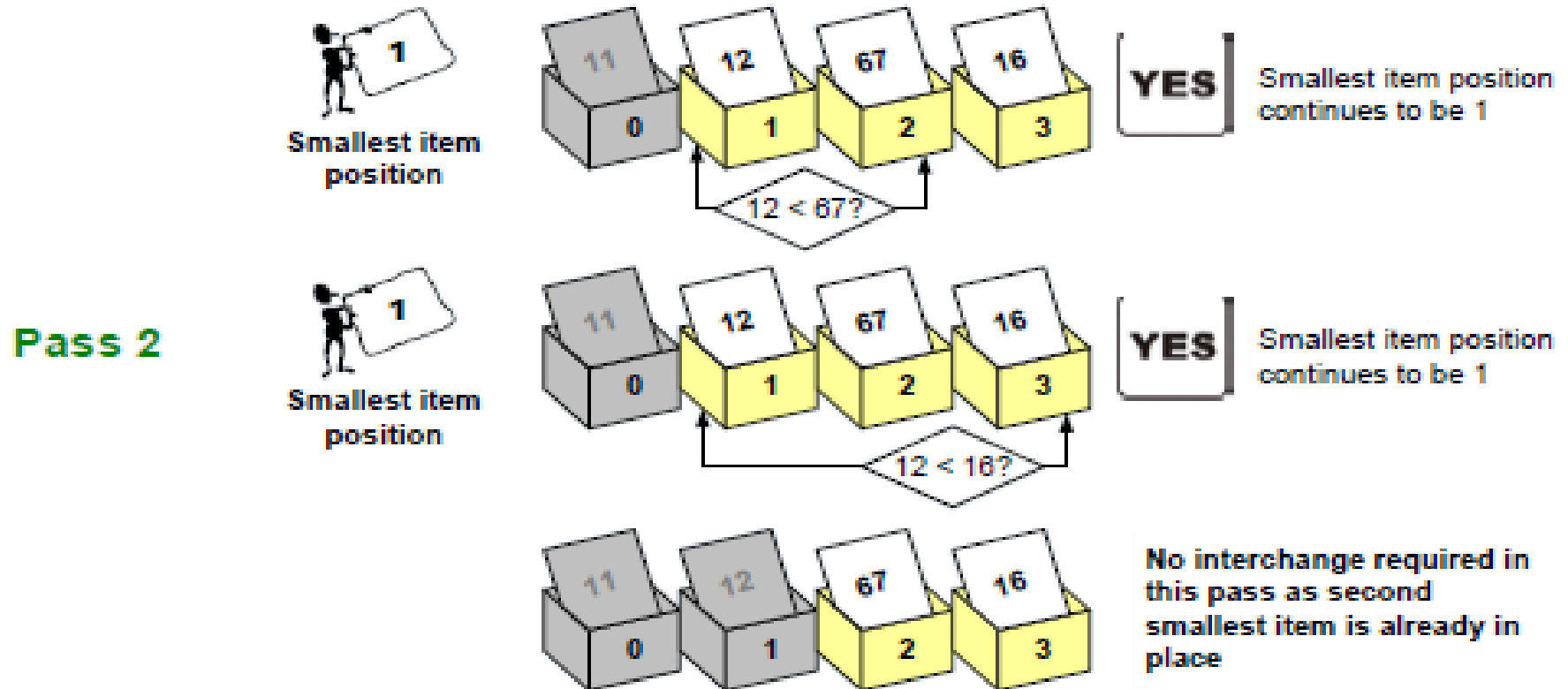
Selection Sort – example
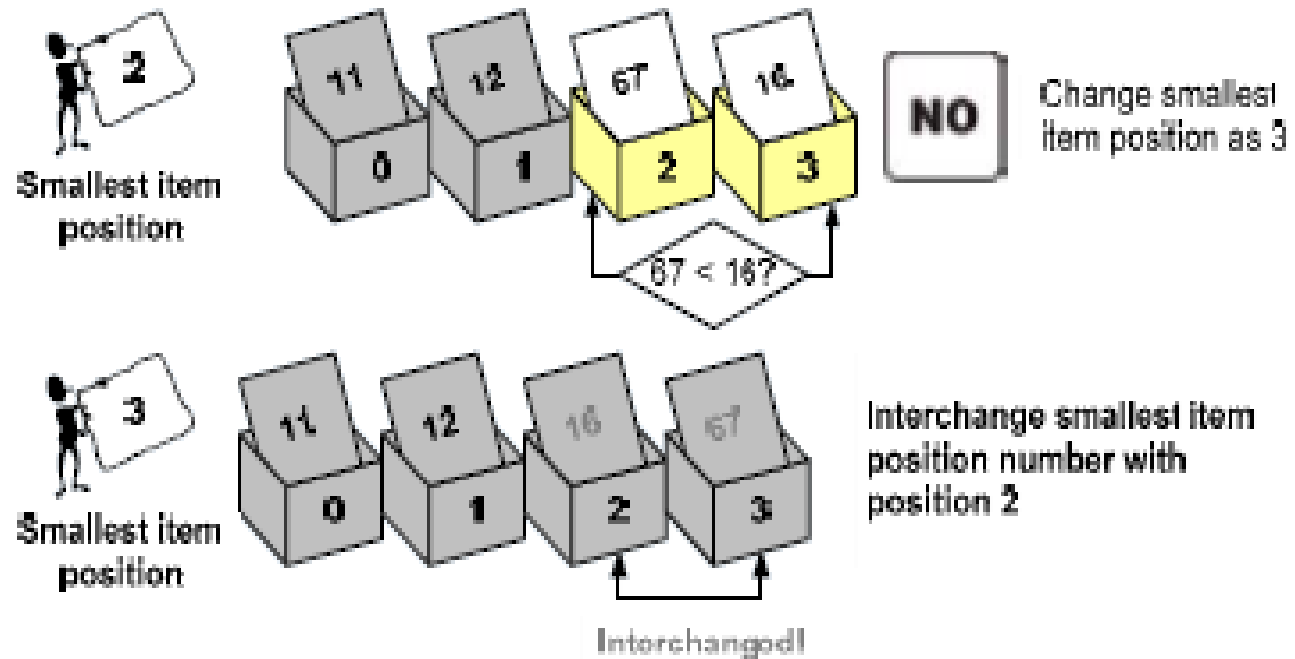
Selection Sort – procedure

for(i = 0; i < n-1; i++) // loop for number of pass

  {   pos = i; small = a[i];

    for(j=i+1; j<n; j++) //loop for searching the smallest

      {   if(small > a[j]) // finding the smallest

         {   pos = j; // pos for interchanging

            small = a[j]; // assigning current small value

        }   }

a[pos] = a[i]; //interchanging values

a[i] = small; }

Selection Sort – illustration
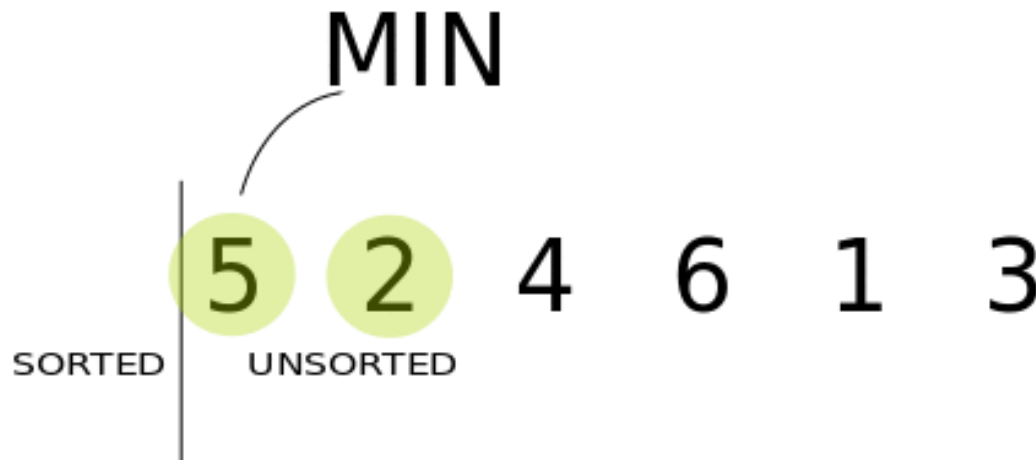
Go to posts/chat box for the link to the question **PQn. S13.2**

submit your solution in next 2 minutes
The session will resume in 3 minutes

Selection Sort for Strings

```c
int main() {
    char arr[100][100], temp[100], minStr[100];
    int n,i, j, min_idx=i;
    printf("enter the number of strings");
    scanf("%d", &n);
    printf("Enter the array\n");
    fflush(stdin);     // clear the buffer
    for (i = 0; i < n; i++)
        gets(arr[i]);
 // Moving boundary of unsorted subarray
    for (i = 0; i < n-1; i++) {
// Find the minimum element in unsorted array
strcpy(minStr, arr[i]);
```

Selection Sort for Strings

```
for (j = i+1; j < n; j++)         {
 // If minStr is greater than arr[j]
if (strcmp(minStr, arr[j]) > 0)         {
// Make arr[j] as minStr and update min_idx
            strcpy(minStr, arr[j]);
            min_idx = j;        }     }
 // Swap the found minimum element with the first element
      if (min_idx != i)        {
strcpy(temp, arr[i]);
//swap item[pos] and item[i]
            strcpy(arr[i], arr[min_idx]);
            strcpy(arr[min_idx], temp); } }
   printf("\nSorted array is\n");
   for (i = 0; i < n; i++)
      puts( arr[i]); } }
```

String Selection Sort input/output

```
Select E:\PSUC_ONLINE\sel.exe

enter the number of strings3
Enter the array
Mango
Banana
Apple

Sorted array is
0: Apple
1: Banana
2: Mango

Process returned 0 (0x0)   execution time : 6.820 s
Press any key to continue.
```

Comparison between Bubble Sort and Selection Sort

Basis for Comparison	Bubble Sort	Selection Sort
Basic	Adjacent element is compared and then swapped	Largest element is selected and swapped with the last element(in case of ascending order)
Efficiency	Inefficient	Improved efficiency when compared to Bubble sort
Method	Exchanging	selection
Speed	Slow	Fast when compared to Bubble sort

Summary

- Selection Sort
- Selection Sort with strings

Summary of S13

- Bubble Sort

- Bubble sort with Strings

- Selection Sort

- Selection Sort with strings

- Comparison of bubble and selection sort