

جامعة المنارة

كلية: الهندسة

قسم: الهندسة المعلوماتية

اسم المقرر: الخوارزميات وبنى المعطيات 2

رقم الجلسة (11)

عنوان الجلسة

خوارزميات الترتيب (الجزء الثاني)



الفصل الدراسي الأول
العام الدراسي 2023-2024

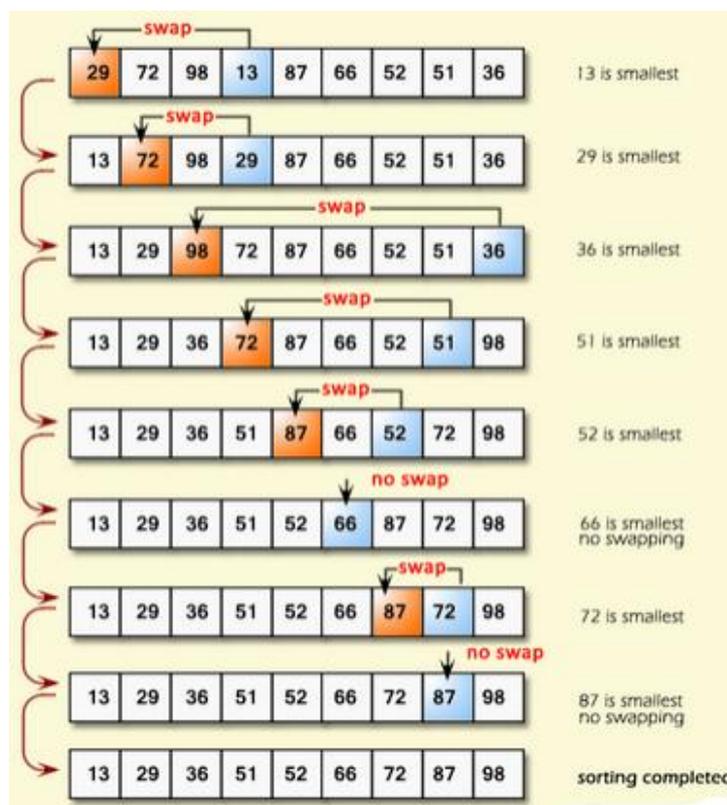
الغاية من الجلسة

- ✓ تطبيق خوارزمية الترتيب بالاختيار.
- ✓ تطبيق خوارزمية الترتيب بالحشر.
- ✓ تطبيق خوارزمية الترتيب السريع.
- ✓ كتابة الأكواد الخاصة بالخوارزميات السابقة.

1- خوارزمية الترتيب بالاختيار Selection Sort Algorithm:

مثال: طبق خوارزمية الترتيب بالاختيار لترتيب المصفوفة التالية:

29,72,98,13,87,66,52,51,36



البرنامج:

```
// C++ program for implementation of selection sort
#include <iostream>
using namespace std;

void selectionSort(int arr[], int n)
{
```

```
int i, j, mini, pos;

for (i = 0; i < n-1; i++)
{
// Find the minimum element in unsorted array
    mini = arr[i]; pos=i;
    for (j = i+1; j < n; j++)
        if (arr[j] < mini)
        {
            mini = arr[j];
            pos=j;
        }

    swap(arr[pos], arr[i]);
}
}

void printArray(int arr[], int size)
{
    int i;
    for (i=0; i < size; i++)
        cout << arr[i] << " ";
    cout << endl;
}

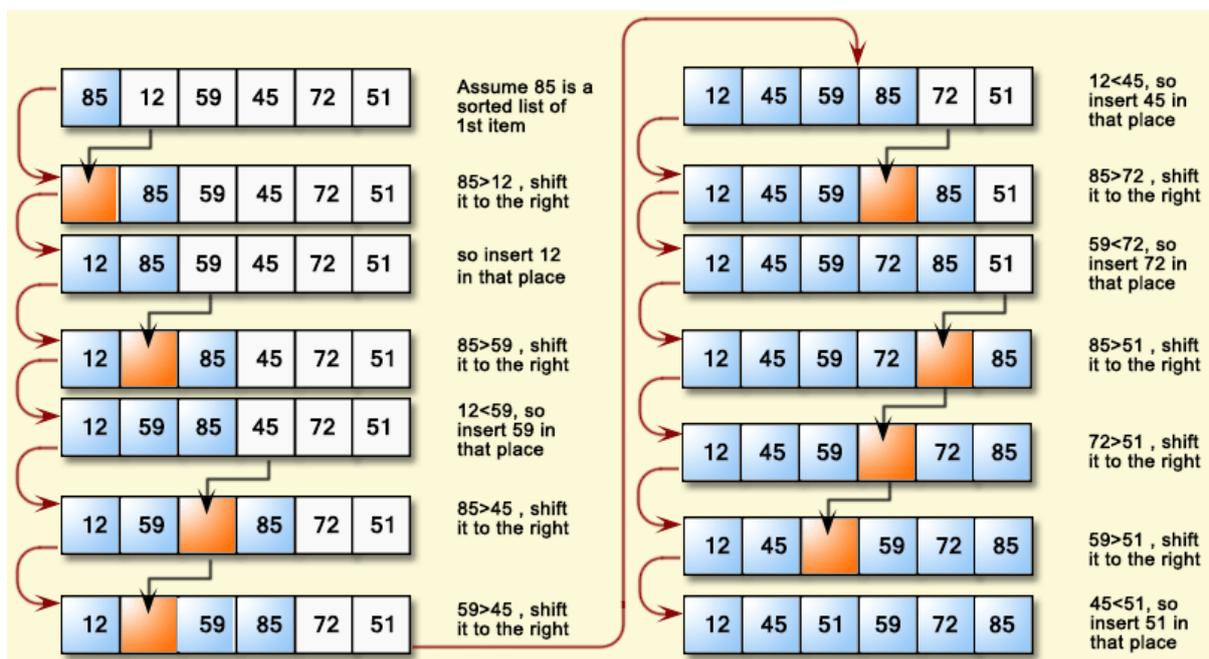
int main()
{
    int n, i;
    cout << "\nEnter the number of data element to be sorted: ";
    int arr[100];
    cin >> n;
    for (i = 0; i < n; i++)
    {
        cout << "Enter element " << i+1 << ": ";
        cin >> arr[i];
    }

    selectionSort(arr, n);
    cout << "Sorted array: \n";
    printArray(arr, n);
    return 0;
}
```

2- خوارزمية الترتيب بالحشر Insertion Sort Algorithm :

مثال : طبق خوارزمية الترتيب بالحشر لترتيب المصفوفة التالية :

85,12,59,45,72,51



: البرنامج

```
// C++ program for insertion sort
#include <iostream>
using namespace std;

/* Function to sort an array using insertion sort*/
void insertionSort(int arr[], int n)
{
    int i, key, j;
    for (i = 1; i < n; i++)
    {
        key = arr[i];
        j = i - 1;

        /* Move elements of arr[0..i-1], that are
        greater than key, to one position ahead
        of their current position */
```

```
while (j >= 0 && arr[j] > key)
{
    arr[j + 1] = arr[j];
    j = j - 1;
}
arr[j + 1] = key;
}
}

// A utility function to print an array of size n
void printArray(int arr[], int n)
{
    int i;
    for (i = 0; i < n; i++)
        cout << arr[i] << " ";
    cout << endl;
}

/* Driver code */
int main()
{
    int n, i;
    cout<<"\nEnter the number of data element to be sorted: ";

    int arr[100];
    cin>>n;

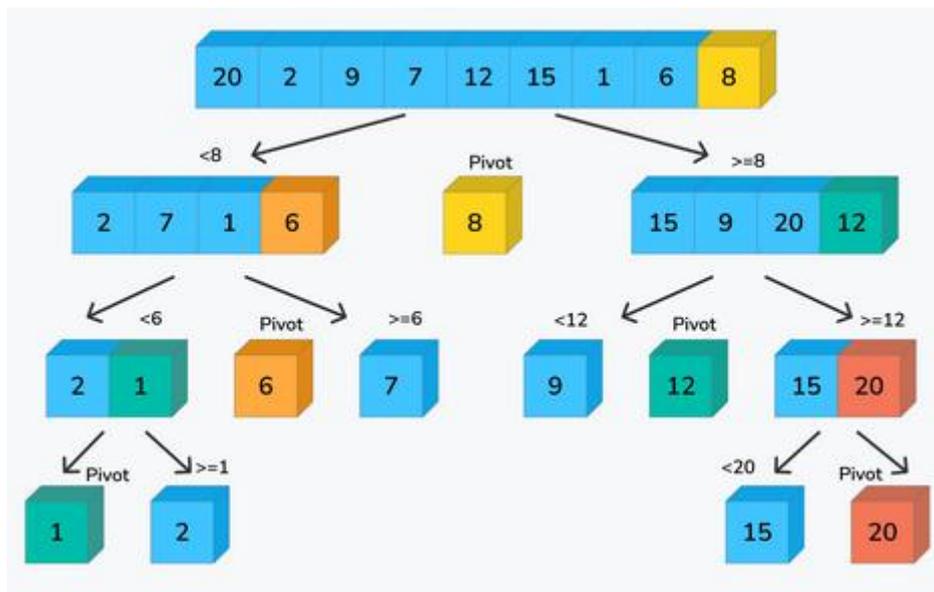
    for(i = 0; i < n; i++)
    {
        cout<<"Enter element "<<i+1<<": ";
        cin>>arr[i];
    }

    cout<<"\nSorted Data\n ";
    insertionSort(arr, n);
    printArray(arr, n);

    return 0;
}
```

3- خوارزمية الترتيب السريع Quick sort algorithm :

مثال : طبق خوارزمية الفرز السريع لترتيب عناصر المصفوفة التالية : 20,2,9,7,12,15,1,6,8



البرنامج :

```
// Quick sort in C++
#include <iostream>
using namespace std;

// function to print the array
void printArray(int array[], int size) {
    int i;
    for (i = 0; i < size; i++)
        cout << array[i] << " ";
    cout << endl;
}

// function to rearrange array (find the partition point)
int partition(int array[], int low, int high) {

    // select the rightmost element as pivot
    int pivot = array[high];

    // pointer for greater element
    int i = low;
    int j = low;
    while( i < high){
```

```
        if(array[i] > pivot){
            i++;
        }
        else{
            swap(array[i],array[j]);
            i++;
            j++;
        }
    }
    swap(array[j],array[high]);
    return j;
}

void quickSort(int array[], int low, int high) {
    if (low < high) {

        // find the pivot element such that
        // elements smaller than pivot are on left of pivot
        // elements greater than pivot are on right of pivot
        int pi = partition(array, low, high);

        // recursive call on the left of pivot
        quickSort(array, low, pi - 1);

        // recursive call on the right of pivot
        quickSort(array, pi + 1, high);
    }
}

// Driver code
int main() {

    int n, i;
    cout<<"\nEnter the number of data element to be sorted: ";
    cin>>n;
    int arr[100];
    for(i = 0; i < n; i++)
    {
        cout<<"Enter element "<<i+1<<": ";
        cin>>arr[i];
    }

    // perform quicksort on data
    quickSort(arr, 0, n - 1);

    cout << "Sorted array in ascending order: \n";
    printArray(arr, n);
}
```

تمرين غير محلول :

طبق كل من خوارزميات الترتيب السابقة لترتيب المصفوفة التالية : 16,3,20,9,12,70,44