Algorithms Analysis & Design

Lecture 1

- Lecturer: Dr. Tarek Morad
- E-mail: hagras@gmail.com
- Course Web Site:

  www.volny.cz/algorithms

- Textbook: *Introduction to Algorithms*, Cormen, Leiserson, Rivest, Stein

  (second edition)

Runtime Sorting Example

```
Sort1()
{
int A[4]= {1,5,3,7},
     i,j,k1;

for( i = 0; i < n-1; i++)
  {
   for( j = i+1; j<n; j++)
      if( A[j] < A[i] )
       { k1 = A[i]; A[i] = A[j]; A[j]=k1;}
  }

}
```

```
Sort2()
{
int A[4]= {1,5,3,7}, F[4]={0,0,0,0},
     i,j,k1;

 for(i=0;i<n-1;i++)
   {
     F[i]=1;
     for(j=0;j<n;j++)
        if(F[j] == 0  && A[j] < A[i])
         {k1=A[i]; A[i]=A[j]; A[j]=k1;}
    }

}
```

Constants Cost Example

```
Sort1()
{
int A[4]= {1,5,3,7},
     i,j,k1;

for( i = 0; i < n-1; i++)
  {
   for( j = i+1; j<n; j++)
      if( A[j] < A[i] )
       { k1 = A[i]; A[i] = A[j]; A[j]=k1;}
  }


}
```

```
Sort3()
{
int A[4]= {1,5,3,7},i,j,k,k1,F,min;
 for(i=0;i<n-1;i++)
   {
     min = A[i];
     for(j=i+1;j<n;j++)
        if(A[j]<min) {min=A[j]; k=j; F=1;}
     if ( F==1)
       { k1 = A[i]; A[i] = A[k]; A[k] = k1;}
   }
}
```

Algorithm

- An algorithm is any well-defined computational procedure that takes some value, or set of values as ***input*** and produce some value or set of values, as ***output***.
- An algorithm is a sequence of computational steps that transform ***input*** into ***output***.

Algorithm Analysis

- Computational Model

  (uniprocessor- Random Access Memory (RAM))

  - All memory equally expensive to access
  - No concurrent operations
  - All reasonable instructions take unit time
    - Except, of course, function calls
  - Constant word size

Input Size

- Number of items
- Number of bits
- Number of nodes and edges

Algorithm Quality Measure

- Predicting the resources
  (*memory, communication band width, hardware*)
- Most often the computational time
  (*running time*)

Algorithm Efficiency

- Computer A
- Insertion Sort ($c_1 n^2$)
- n=1,000,000
- 10^9 instruction/sec
- C_1=2

$$\frac{2(10^6)^2 \quad instructions}{10^9 \quad instruction/\sec}$$

- 2000 sec

- Computer B
- Merge Sort ($c_2 nlgn$)
- n=1,000,000
- 10^7 instruction/sec
- C_2=50

$$\frac{50(10^6 \lg 10^6) \quad instructions}{10^7 \quad instruction/\sec}$$

- 100 sec