Εργαστηριακές Σημειώσεις – Εβδομάδα 7

Σε αυτό το εργαστήριο θα μελετηθούν:

- *Η βιβλιοθήκη time.h*
- *Χρησιμοποίηση της βιβλιοθήκης για καταγραφή του χρόνου εκτέλεσης μιας συνάρτησης*

Importing the library

- #include <time.h>
- The variables
  - typedef long clock_t // used for storing processor time

The variables

- clock_t start;
  - For recording the start time of the application
- clock_t finish;
  - For recording the finish time of the application
- double duration;
  - duration is the program time execution
  - duration = finish – start;

Duration

- Duration returns the number of Clock Cycles
- To convert to seconds
  - duration = (double)(finish - start) / CLOCKS_PER_SEC;

Printing the results

- printf("Start:%u\n", start);
- printf("Finish:%u\n", finish);
- printf("Duration:%5.1f seconds\n", duration);
- printf("Duration:%5.0f miliseconds\n", duration*1000);

Application flow

Initialize program variables

start = clock();

Execution of the method we want to time

finish = clock();

Print the results

Time a real scenario

- Linear Search
- Binary Search

Linear Search

- *Δεδομένα Εισόδου*: Πίνακας X με n στοιχεία, ταξινομημένος από το μικρότερο στο μεγαλύτερο, και ακέραιος k.
- *Στόχος*: Να εξακριβώσουμε αν το k είναι στοιχείο του X.
- *Γραμμική Διερεύνηση*: εξερευνούμε τον πίνακα από τα αριστερά στα δεξιά.

```
int linear( int X[], int n, int k){
 int i=0;
 while ( i < n ) {
   if (X[i] == k) return i;
   if (X[i] > k)  return -1;
   i++;
 }
 return -1;
}
```

Binary Search

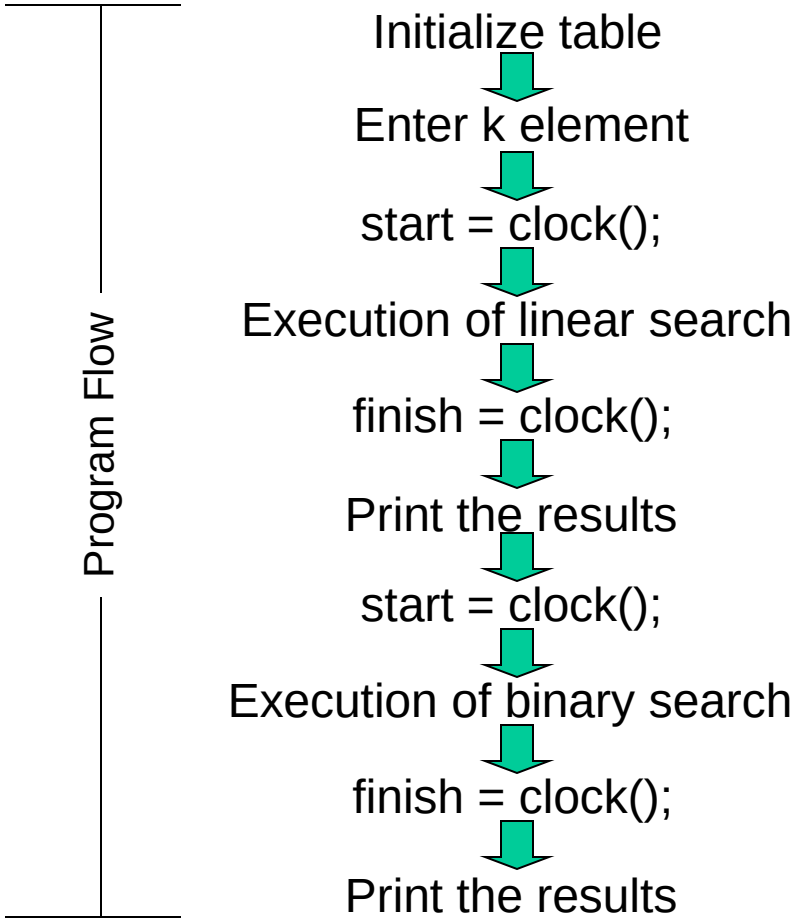
- *Δυαδική Διερεύνηση:* βρίσκουμε το μέσο του πίνακα και αποφασίζουμε αν το k ανήκει στο δεξιό ή το αριστερό μισό. Επαναλαμβάνουμε την ίδια διαδικασία στο "μισό" που μας ενδιαφέρει.

```
int binary( int X[],int n,int k){
 int low = 0, high = n-1;
 int mid;
 while ( low < high ){
   mid = (high + low)/2;
   if (X[mid] < k) low = mid + 1;
   else
   if (X[mid] > k) high=mid-1;
   else return mid;
  }
 return -1;
}
```

Let's start the implementation

- Define a variable MAX=1000000
- Create a table of size MAX
- Initialize table from 0-(MAX-1)
- Implement both linear and binary search
- Create a menu for entering the k-element

Program Flow



Tryouts

- k=
  - 1000000
  - 500000
  - 2
  - Change MAX to 1.000.000 or more!

- Compare and explain the results

The End