Universidad Nacional de Río Cuarto
Facultad de Ciencias Exactas, Físico-Químicas y Naturales
Departamento de Computación
Estructura de Datos yAlgoritmos – Algoritmos I – Estructura de la Información
Agosto de 2001

Trabajo Práctico Nº 1 (Segunda Parte) : Tiempo de Ejecución e Introducción a C++

1. Obtener la forma Cerrada y Velocidad de Crecimiento de los siguientes algoritmos.

El procedimiento *intercambia* es de O(1).

a) procedure *Inserción* **(var** A:array[1..n] of integer, n:integer**)**

```
var i,j: integer;
begin
        A[0]:= -∞;
        for i:2 to n do begin
                j:=i;
                while A[j]<A[j-1] do begin
                        intercambia(A[j],A[j-1]);
                        j:=j-1;
                end;
        end;
end
```

b) procedure *Selección*(**var** A:array[1..n] of integer, n:integer)

```
var clave_menor, indice_menor,i,j: integer;
begin
        for i:1 to n-1 do begin
                indice_menor:=i;
                clave_menor:=A[i];
                for j:i+1 to n do begin
                        if A[j]<clave_menor then begin
                                clave_menor:=A[j];
                                indice_menor:=j;
                        end;

            intercambia(A[i],A[indice_menor]);
        end;
    end;
end
```

c) Procedure *Combinar_listas_Ordenadas* (**var** A: array [1..n] of integer, n:integer, B:array[1..m] of integer, C: array[1..n+m] of integer)

```
var  indexA, indexB, indexC;
begin
        indexA:=1;
        indexB:=1;
        indexC:=1;
While indexA≤n and indexB≤n do begin
        if A[indexA] < B[indexB] then
                begin
                  C[indexC] := A[indexA];
```

```
                    indexA:=indexA+1;
                end;
            else
                begin
                  C[indexC] := B[indexB];
                  indexB:= indexB +1;
                end;

            indexC := indexC +1;
    end;  {while}

    if indexA>n then
            Mover (B[indexB], ... , B[m] to C[indexC], ... , C[n+m]);
    else
            Mover (A[indexA], ... , A[n] to C[indexC], ... , C[n+m]);
end.
```

d) funcion *potencia*(x,n:integer):integer;

```
    begin
            if n=0 then
                    potencia:=1;
            else
                    if n=1 then
                            potencia:=x;
                    else
                            if even(n) then
                                    potencia:=potencia(x*x,n div2);
                            else
                                    potencia:=potencia(x*x,n div2)*x;
    end
```

Suponga que *concat* consume tiempo constante.

e) funcion *prims*(n:natural):listanat

```
    var s1: listanat;
    begin
            if n=0 then
                    prims:=[0];
            else
                    begin
                            s1:=prims(n-1);
                            prims:=concat(s1,[n]);
                    end
```

2. Traduzca los algoritmos de los incisos anteriores a C++.

3. Desarrolle en **C++** los siguientes algoritmos. Obtener también su Forma Cerrada y Velocidad de Crecimiento:

- Dado un arreglo de enteros, retorne el mínimo y el máximo elemento del arreglo.
- Busqueda Dicotómica.