

Algoritmos y Estructuras de Datos II – 18 de diciembre de 2017
Examen Final Teórico-Práctico

Alumno: Email:
Siempre se debe explicar la solución, una respuesta correcta no es suficiente sino viene acompañada de una justificación que demuestre que la misma ha sido comprendida. Las explicaciones deben ser completas. La utilización de código con un nivel de abstracción excesivamente bajo influirá negativamente. En los ejercicios con varios incisos, por favor, no resolver varios incisos simultáneamente, sino cada uno por separado (pero no hace falta que sea en hojas aparte). Pero que cada ejercicio se vaya en hoja separada.

1. Para cada una de los siguientes algoritmos determinar **por separado** cada uno de los siguientes incisos.

```
fun f(i : Int) ret b : bool
  b := (i mod 2) == 0;
end fun
```

```
proc g(in/out a : array[1..n] of Int , in i : nat , in j : nat)
  var x : Int;
  x := a[i];
  a[j] := a[i];
  a[i] := x;
end proc
```

```
fun h(a : array[1..n] of Int , i : nat , j : nat) ret m : Int
  m := a[i];
  for k := i+1 to j do
    if a[k] < m then m := a[k]
  fi
od
end fun
```

```
proc p(in/out a : array[1..n] of Int , out k : nat)
  k := 0;
  for i := 1 to n do
    if not f(a[i]) then
      for j := i+1 to n do
        if f(a[j]) then
          g(a,i,j);
          k := i+1;
        fi
      od
    fi
  od
end proc
```

(a) ¿Qué hace?

(b) ¿Cómo lo hace?

(c) El orden del algoritmo.

(d) Utilizar nombres adecuados para los identificadores de procedimientos y funciones.

```
proc q(in/out a : array[1..n] of Int)
  var k,r : nat;
  p(a,k);
  for i := 1 to k do
    r := h(a,i,k);
    g(a,i,r);
  od
  for i := k+1 to n do
    r := h(a,i,n);
    g(a,i,r);
  od
end proc
```

2. En una ciudad de Irlanda el transporte público se paga con la tarjeta Red Irish. En la tarjeta se conservan el saldo (en centavos, un natural), la hora del último viaje (en minutos después del Big Bang, un natural) y la línea de colectivos en que se realizó el último viaje (otro natural). Se pide especificar el TAD tarjeta con sus constructores Comprar, Cargar, Viajar y las funciones:

viveAIDfa que, dada una tarjeta, dice si el saldo nunca superó los 20000 centavos.

UltimoBondi que da la última línea en que se viajó, si es que la hay.

UltimaHora que da la hora del último viaje (de haberlo).

Nótese que:

(a) Al comprar una tarjeta, el saldo es cero y todavía no se realizó ningún viaje.

(b) Al cargar una tarjeta, se carga una cantidad positiva de centavos, cargar una tarjeta no cambia el último viaje realizado.

(c) Si el último viaje fue realizado hace menos de 60 minutos, se puede realizar gratuitamente un segundo viaje siempre que sea en otra línea de colectivo. Pero si luego se hace un tercer viaje, debe pagarse 1538 centavos por el tercero, por más que sea realizado dentro de los 60 minutos (y por lo tanto debe contarse con saldo no menor a 1538 centavos).

(d) Si no hay un último viaje, o si el último viaje no fue realizado hace menos de 60 minutos, o fue realizado hace menos de 60 minutos pero en la misma línea de colectivo, el saldo debe de ser no menor a 1538 centavos y el viaje se paga 1538 centavos.

3. Te dedicás a criar pececitos de colores. Para eso trasladás una pareja de reproductores a una pecera de cría para que hagan lo suyo, y los huevos eclosionen y los pecesitos crezcan sin que se los coman los demás peces. Tenés una sola pecera de cría. Para cada hembra sabés en qué fecha es capaz de poner huevos y en qué fecha sus hijos ya están suficientemente grandes como para dejar la pecera de cría y vivir entre peces adultos. Escribí un algoritmo en pseudocódigo que dados dos arreglos de tamaño N : $D[i]$ y $A[i]$ con los tiempos de desove y llegada a la adultez de las hembras y sus (potenciales) hijos, dé como salida un arreglo de booleanos de tamaño N que diga si cada hembra pondrá sus huevos en la pecera de cría o no. La idea es que la mayor cantidad de hembras se reproduzcan y esto se puede resolver con un algoritmo voraz (como los peces). Explicá el criterio elegido para la reproducción (de los peces).

4. El Problema de Hacer la Cola

Estás de patovica en un súper y es la hora de cerrar. Hay una sola cola con n clientes y cada cliente tiene C_i objetos cargados en su chango. Los dueños del súper deciden habilitar otra caja y la persona de la caja uno tickea diez objetos por minuto, pero la de la caja dos es más lenta y solamente alcanza a tickear seis productos por minuto. Se pide asignar los clientes a dos colas, una por cada caja tratando de minimizar el tiempo hasta que todos los clientes sean atendidos respetando el orden de la cola original. Es decir, ningún cliente puede quedar en la nueva cola por delante de alguien que haya estado antes en la cola original; sí vale que lo atiendan antes si están en colas distintas.

5. (Para alumnos libres) Explicá el algoritmo de Kruskal: qué problema resuelve, sus entradas y salidas y cómo funciona.