

# Algoritmos y Estructuras de Datos

## Primer Parcial – Viernes 9 de mayo de 2025

Libreta	Apellido y Nombre	E1	E2	E3	E4	Nota Final	Corrigió

- Es posible tener una hoja (2 carillas), escrita a mano, con los anotaciones que se deseen, más los dos apuntes del campus
- Cada ejercicio debe entregarse en **hojas separadas**
- Incluir en cada hoja el número de libreta, número de hoja, apellido y nombre
- El parcial se aprueba con 60 puntos y al menos 2 preguntas teóricas correctas

### E1. TADs y especificación de problemas [40 pts]

- 1) La Panadería *El Progreso* recibe todos los días panes, facturas, masas, y otras delicias para venderla a sus clientes. Una vez que cierra sus puertas todo lo que no se vendió se envía a un comedero del barrio, por lo que todos los días se empieza con un stock completamente nuevo.  
Los clientes son atendidos por estricto orden de llegada y realizan su pedido a algún empleado libre. Si no hay mercadería suficiente disponible para cumplir su pedido se retiran con las manos vacías. Caso contrario se le entrega y luego esperan para pagar, también en orden de llegada. Una vez que pagaron se retiran a disfrutar sus panificados.  
Nos piden modelar en un TAD el funcionamiento de El Progreso según esta descripción, teniendo en cuenta que nos importa en todo momento saber cuántos clientes están esperando que los atiendan y cuántos están esperando para pagar.
  - a) Indique las operaciones (procs) del TAD con todos sus parámetros y los renombres de tipo que considere necesarios.
  - b) Describa el TAD en forma completa, indicando sus observadores, los requiere y asegura de las operaciones. Puede agregar los predicados y funciones auxiliares que necesite, con su correspondiente definición.
  - c) Cuando presentamos nuestro tad nos dicen que ahora quieren saber qué empleado atiende más clientes cada día. ¿Debería modificar su TAD para reflejar esto? ¿Cómo? Responda en palabras, en forma breve y precisa.
- 2) Suponiendo que no cuenta con la operación *setKey*, complete la especificación de la operación **definir** en el siguiente TAD

```

TAD Diccionario < K, V > {
  obs data: dict<K,V>
  proc nuevoDiccionario () : Diccionario<K,V> {
    asegura {res.data = <>}
  }
  // define la clave en el diccionario, la agrega si no existía
  proc definir (inout d: Diccionario<K,V>, in k: K, in v: V) {
    ...
  }
}

```

### E2. Preguntas teóricas [10 pts]

Responder las siguientes preguntas **sin justificar** su respuesta.

- 1) ¿Qué programas *S* hacen válida la tripla de Hoare  $\{True\} S \{True\}$ ?
- 2) Si el cuerpo de un ciclo es un condicional y en el *then* la función variante propuesta se reduce estrictamente pero en el *else* puede quedar inalterada cuando se lo ejecuta, ¿Qué sucedería al aplicar el teorema del variante? ¿Cómo se puede arreglar?
  - a) El ciclo termina y no hay nada que arreglar.
  - b) El ciclo no termina, y hay que agregar un decremento de la función variante.
  - c) El ciclo podría no terminar y se debería modificar el *else* para que se reduzca la función variante.
  - d) El ciclo podría no terminar, hay que modificar la guarda del ciclo.
- 3) ¿Puede ocurrir que un invariante *I* satisfaga los puntos 1 y 3 del teorema del invariante, valga cada vez que termina de ejecutar **S** (el cuerpo del ciclo) pero no satisfaga el punto 2?
  - a) Sí
  - b) No
  - c) Depende de la precondition del ciclo
  - d) Depende de la postcondición del ciclo

- 4) Si un invariante  $I$  cumple los puntos 1 y 2 del Teorema del Invariante pero no el 3, ¿qué tiene más sentido?
- Debilitarlo
  - Ni debilitarlo ni fortalecerlo, cambiarlo por uno diferente.
  - Fortalecerlo
  - Debilitarlo en la precondition y fortalecerlo en la postcondición.
- 5) ¿Por qué la garantía del teorema del invariante no se expresa como tripla de Hoare y se habla de correctitud parcial?
- Porque los ciclos no terminan.
  - Porque nos alcanza con la función variante.
  - Porque no sabemos cuántas veces se va a ejecutar.
  - Porque no tenemos una cota a la cantidad de veces que se va a ejecutar.

### E3. Precondición más débil [20 pts]

Para los siguientes algoritmos **S** con sus post condiciones  $Q$ , indique cuál de las precondiciones propuestas es la *Precondición más débil* y justifique muy brevemente en palabras.

1) **S**  $\equiv$

```

if (a mod 2 = 0)
  a := |a| + 1
else
  a := |a| * 2
endif

```

$Q \equiv \{a \bmod 2 = 0\}$

- $P \equiv \{a \bmod 2 = 0\}$
- $P \equiv \{a \bmod 2 \neq 0\}$
- $P \equiv \{True\}$

2) **S**  $\equiv$

```

j := i - 2
s[j] := 2 * i

```

$Q \equiv \{(\forall k : \mathbb{Z})(0 \leq k < |s| \rightarrow_L s[k] \bmod 2 = 0)\}$

- $P \equiv \{(\forall k : \mathbb{Z})(0 \leq k < |s| \wedge k \neq i - 2 \rightarrow_L s[k] \bmod 2 = 0)\}$
- $P \equiv \{2 \leq i < |s| \wedge (\forall k : \mathbb{Z})(0 \leq k < |s| \wedge k \neq i - 2 \rightarrow_L s[k] \bmod 2 = 0)\}$
- $P \equiv \{i \bmod 2 = 0 \wedge 2 \leq i < |s| \wedge (\forall k : \mathbb{Z})(0 \leq k < |s| \wedge k \neq i - 2 \rightarrow_L s[k] \bmod 2 = 0)\}$

3) **S**  $\equiv$

```

if (x > y)
  y := x
else
  y := 3
endif

```

$Q \equiv \{y > 0\}$

- $P \equiv \{x > y\}$
- $P \equiv \{x > y \vee x \leq y\}$
- $P \equiv \{(x > 0 \wedge x > y) \vee (x \leq y)\}$

4) **S**  $\equiv$

```

if (i mod 2 = 0)
  s[i] := 1
else
  s[i] := 5
endif

```

$Q \equiv \{(\forall i : \mathbb{Z})(0 \leq i < |s| \wedge i \bmod 2 \rightarrow_L s[i] = 1)\}$

- $P \equiv \{0 \leq i < |s| \wedge i \bmod 2 = 0 \wedge (\forall j : \mathbb{Z})(0 \leq j < |s| \wedge j \bmod 2 = 0 \wedge j \neq i \rightarrow_L s[j] = 1)\}$
- $P \equiv \{0 \leq i < |s| \wedge i \bmod 2 = 0 \wedge (\forall j : \mathbb{Z})(0 \leq j < |s| \wedge j \neq i \rightarrow_L s[j] = 1)\}$
- $P \equiv \{i \bmod 2 = 0 \wedge (\forall j : \mathbb{Z})(0 \leq j < |s| \wedge j \bmod 2 = 0 \wedge j \neq i \rightarrow_L s[j] = 1)\}$

### E4. Correctitud del ciclo [30 pts]

Dado el siguiente programa con su especificación

$P_c \equiv \{n > 0 \wedge res = 1 \wedge i = 1\}$

```

while (i < n)
  res := res * i
  i := i + 1;
endwhile

```

$Q_c \equiv \{res = (n - 1)!\}$

- Escriba el Invariante del ciclo
- Demuestre formalmente que el invariante propuesto cumple los axiomas del Teorema del Invariante
- Decida si las siguiente funciones pueden o no usarse como funciones variantes para demostrar la terminación del ciclo y justifique muy brevemente por qué.
  - $f_v = n$
  - $f_v = n - i$
  - $f_v = n - 1 - i$
  - $f_v = i - n$
  - $f_v = n! - res$