

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA
ESTRUCTURA DE DATOS Y PROGRAMACIÓN METÓDICA
2da práctica (tipo a)
(Segundo semestre de 2017)

Horario 0581: prof. V.Khlebnikov

Horario 0582: prof. A.Bello R.

Duración: 1 hora 50 min.

Nota: No se puede usar ningún material de consulta.

La presentación, la ortografía y la gramática influirán en la calificación.

Puntaje total: 20 puntos

Pregunta 1 (5 puntos - 25 min.) Cuando las constructoras del TAD Lista son las operaciones de la lista vacía, la lista unitaria y concatenación, se puede construir la siguiente lista: $[\alpha] ++ ([\varphi] ++ ([\eta] ++ ((([\rho] ++ ([\eta] ++ [\mu])) ++ [\varepsilon]) ++ [\nu]) ++ [\eta])))$. Se define la operación $\theta\varepsilon\sigma\eta$ con el perfil $\theta\varepsilon\sigma\eta : \text{elemento lista} \rightarrow \text{ent}$ y con las siguientes ecuaciones:

- (1) $\theta\varepsilon\sigma\eta(e, x) = -1 \Leftarrow \neg \text{está?}(e, x)$
- (2) $\theta\varepsilon\sigma\eta(e, x) = 0 \Leftarrow \neg \text{es-lista-vacía?}(x) \wedge \text{izquierdo}(x) == e$
- (3) $\theta\varepsilon\sigma\eta(e, x) = 1 + \theta\varepsilon\sigma\eta(e, \text{elim-izq}(x)) \Leftarrow \neg \text{es-lista-vacía?}(x) \wedge \text{está?}(e, \text{elim-izq}(x)) \wedge \text{izquierdo}(x) \neq e$

a) (1,5 puntos - 8 min.) Perfectamente conocemos cómo se hace el cálculo de, por ejemplo, $(5 + (9 - 3) * 8 + 2) * 2$. Pero presente el cálculo de $\theta\varepsilon\sigma\eta(\eta, [\alpha] ++ ([\varphi] ++ ([\eta] ++ ((([\rho] ++ ([\eta] ++ [\mu])) ++ [\varepsilon]) ++ [\nu]) ++ [\eta])))$ en la siguiente forma donde cada expresión está en una línea separada y, en la línea entre estas, se indica el número de la ecuación aplicada:

$\theta\varepsilon\sigma\eta(\eta, [\alpha] ++ ([\varphi] ++ ([\eta] ++ ((([\rho] ++ ([\eta] ++ [\mu])) ++ [\varepsilon]) ++ [\nu]) ++ [\eta])))$
 $\equiv <\text{según la ecuación N}>$
 \dots
 $\equiv <\text{según la ecuación N}>$
 \dots

Si la 1ra ecuación se mantiene y la 3ra ecuación se cambia de la siguiente manera:

- (3) $\theta\varepsilon\sigma\eta(e, x) = 1 + \theta\varepsilon\sigma\eta(e, \text{elim-izq}(x)) \Leftarrow \neg \text{es-lista-vacía?}(x) \wedge \text{está?}(e, \text{elim-izq}(x))$

b) (1,5 puntos - 8 min.) ¿Cuál debe ser la 2da ecuación?

c) (1 punto - 5 min.) Con las nuevas ecuaciones presente el cálculo de $\theta\varepsilon\sigma\eta(\rho, [\alpha] ++ ([\varphi] ++ ([\eta] ++ ((([\rho] ++ ([\eta] ++ [\mu])) ++ [\varepsilon]) ++ [\nu]) ++ [\eta])))$ de la misma forma usada en la respuesta a la pregunta **1a)**.

d) (1 punto - 5 min.) Con las nuevas ecuaciones presente el cálculo de $\theta\varepsilon\sigma\eta(\eta, [\alpha] ++ ([\varphi] ++ ([\eta] ++ ((([\rho] ++ ([\eta] ++ [\mu])) ++ [\varepsilon]) ++ [\nu]) ++ [\eta])))$ de la misma forma usada en la respuesta a la pregunta **1a)**.

Pregunta 2 (5 puntos - 25 min.) Se ha implementado el TAD Pilas, en Python. Pero usted no sabe cómo se ha implementado, lo único que conoce es su interfaz, donde las operaciones se han codificado como métodos de la clase **Stack**. Los métodos son: **isEmpty**, devuelve **True** si la pila está vacía o **False** en caso contrario; **peek**, devuelve la cima de la pila; **pop**, retira y devuelve la cima de la pila y **push**, agrega un elemento a la pila.

Escriba en Python una función que calcule el Fibonacci de un número, empleando una pila. No puede emplear funciones auxiliares ni otro TAD.

Por ejemplo, si en la función `main()` es:

```
def main():
    lista = []
    for i in range(7):
        lista.append(fibonacci(i))
    print(lista)
```

La salida sería:

```
[0, 1, 1, 2, 3, 5, 8]
```

Pregunta 3 (5 puntos - 25 min.) Se presenta el siguiente código:

```
class SecondChance:
    def __init__(self, maxSize):
        self._N = maxSize
        self._count = 0
        self._front = 0
        self._back = self._N - 1
        self._current = self._front
        self._qArray = Array(self._N)
        self._uArray = Array(self._N)

    def isEmpty(self):
        return self._count == 0

    def isFull(self):
        return self._count == self._N

    def __len__(self):
        return self._count

    def _pos(self, item):
        if self.isEmpty():
            return None
        i = self._front; n = self._count
        while self._qArray[i] != item and n > 0:
            n -= 1; i = (i + 1) % self._N
        if n == 0:
            return None
        else:
            return i

    def used(self, item):
        p = self._pos(item)
        if p == None:
            return False
        self._uArray[p] = True
        return True

    def _used_(self, item):
        return self._uArray[self._pos(item)]
```

```

def unused(self):
    if self.isEmpty():
        return False
    i = self._front; n = self._count
    while n > 0:
        self._uArray[i] = False
        n -= 1; i = (i + 1) % self._N
    return True

def enqueue(self, item):
    if not self.isFull():
        self._back = (self._back + 1) % self._N
        self._qArray[self._back] = item
        self._uArray[self._back] = True
        self._count += 1
    else:
        while True:
            if not self._uArray[self._current]:
                self._qArray[self._current] = item
                self._uArray[self._current] = True
                self._current = (self._current + 1) % self._N
                return True
            else:
                self._uArray[self._current] = False
                self._current = (self._current + 1) % self._N

```

Se realizan las siguientes operaciones:

```

q = SecondChance(4)
q.enqueue(11)
q.enqueue(12)
q.enqueue(13)
q.enqueue(14)
q.unused()
q.used(11)
q.used(13)
q.enqueue(15)

```

a) (1 punto - 5 min.) ¿Cuál es el estado de la estructura q en este punto de ejecución?

```
q.enqueue(17)
```

b) (1 punto - 5 min.) ¿Cuál es el estado de la estructura q en este punto de ejecución?

```
q.enqueue(19)
```

c) (1 punto - 5 min.) ¿Cuál es el estado de la estructura q en este punto de ejecución?

```
q.enqueue(21)
```

d) (1 punto - 5 min.) ¿Cuál es el estado de la estructura q en este punto de ejecución?

```
q.enqueue(23)
```

e) (1 punto - 5 min.) ¿Cuál es el estado de la estructura q en este punto de ejecución?

Pregunta 4 (5 puntos - 25 min.) Considere la siguiente implementación de árbol binario:

```
from myarray import *
class BinaryTree:
    def __init__(self, size=100):
        self.root = Array(size)
        self.left = Array(size)
        self.right = Array(size)
        self.index = 1
        for i in range(len(self.root)):
            self.root[i], self.left[i], self.right[i] = None, None, None

    def getRaiz(self):
        return self.root[0]

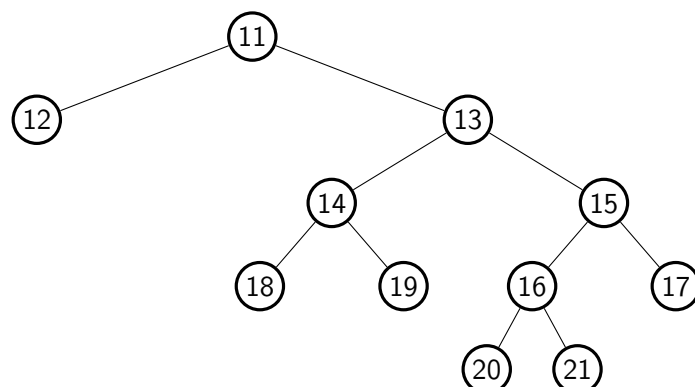
    def setRaiz(self, root):
        self.root[0] = root

    def setElements(self, root, left, right = None):
        assert (self.index - 2) <= len(self.root), "No hay espacio suficiente"
        i = search(root, self.root)
        assert i is not None, "Raíz no existe"
        if left is not None:
            self.root[self.index] = left
            self.left[i] = self.index
            self.index += 1
        else:
            pass
        if right is not None:
            self.root[self.index] = right
            self.right[i] = self.index
            self.index += 1
```

Donde la función `search` está definida fuera de la clase y su código es el siguiente:

```
def search(elem, array):
    for i in range(len(array)):
        if array[i] == elem:
            return i
    return None
```

- a) (1 punto - 5 min.) ¿Cuál sería la secuencia de instrucciones en Python para obtener la representación del siguiente árbol:



- b) (0,5 puntos - 2,5 min.) Agregue el método `size` que devuelve el tamaño del árbol.
- c) (1,5 puntos - 7,5 min.) Agregue el método `leaves` que devuelve en una lista las hojas del árbol.
- d) (2 puntos - 10 min.) Agregue el método `siblings` que devuelve en una lista de pares ordenados, los nodos que son hermanos.



Profesores del curso: V.Khlebnikov
A.Bello R.

La práctica fue preparada por AB(2,4) y VK(1,3)
con L^AT_EX en Linux Mint 18.2 Sonya

Pando, 21 de septiembre de 2017