Algoritmos y Estructuras de Datos II – 23 de julio de 2014
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 c o de nivel de abstracción excesivamente bajo influirá negativamente. En los ejercicios con varios incisos, por favor, no resuelvas varios incisos simultáneamente, sino cada uno por separado (pero no hace falta que sea en hojas aparte).

1. El TAD calc_cola, define una calculadora a cola de la siguiente manera:

TAD calc_cola
constructores
    vacía : calc_cola
    encolar : calc_cola × entero → calc_cola
operaciones
    es_vacía : calc_cola → booleano
    primero : calc_cola → entero   {sólo se aplica a colas no vacías}
    decolar : calc_cola → calc_cola   {sólo se aplica a colas no vacías}
    recolar : calc_cola → calc_cola   {sólo se aplica a colas no vacías}
    opuesto : calc_cola → calc_cola   {sólo se aplica a colas no vacías}
    mas : calc_cola → calc_cola   {sólo se aplica a colas con al menos dos elementos}
    tamaño : calc_cola → nat
    suma : nat × calc_cola → calc_cola   {solo se aplica a pares (n, q) con n menor o igual al tamaño de q}
ecuaciones
    es_vacía(vacía) = verdadero
    es_vacía(encolar(q,i)) = falso

    primero(encolar(vacía,i)) = i
    primero(encolar(encolar(q,j),i)) = primero(encolar(q,j))

    decolar(encolar(vacía,i)) = vacía
    decolar(encolar(encolar(q,j),i)) = encolar(decolar(encolar(q,j)),i)

    recolar(q) = encolar(decolar(q),primero(q))
    opuesto(q) = encolar(decolar(q),-primero(q))
    mas(q) = encolar(decolar(decolar(q)),primero(q)+primero(decolar(q)))
    menos(q) = encolar(decolar(decolar(q)),primero(q)-primero(decolar(q)))

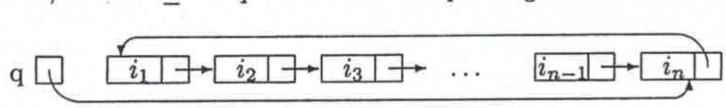
    tamaño(vacía) = 0
    tamaño(encolar(q,i)) = 1 + tamaño(q)

    suma(n,q) = suma_aux(n,0,q)
    suma_aux(0,r,q) = encolar(q,r)
    $n \geq 1 \Longrightarrow$ suma_aux(n,r,q) = suma_aux(n-1,r+primero(q),decolar(q))

Implementá el TAD calc_cola con una lista enlazada circular de enteros, con puntero al último. Es decir, se define

type queue_calc = **pointer to** node
type node = **tuple**
        value: **int**
        next: **pointer to** node
    end

Con esta representación, una calc_cola q de tamaño n se puede graficar como



donde los i_k son números enteros y el primer elemento de la calc_cola q es i_1.

La implementación de las operaciones encolar, decolar, recolar, opuesto, mas y suma deben modificar la calc_cola que reciben como argumento.

2. Sean Y y Z los siguientes procedimientos:

```
proc Y(out m:array[0..n,0..n] of int)
    for i:= 0 to n do
        for j:= 0 to n do
            m[i,j]:= j-i
        od
    od
end fun
```

```
proc Z(out m:array[0..n,0..n] of int)
    for i:= -n to n do
        for j:= máx(0,i) to mín(n,i+n) do
            m[j-i,j]:= i
        od
    od
end fun
```

a) Explicá qué hace cada uno de ellos.

b) Explicá cómo realiza su tarea cada uno de ellos.

c) Calculá los órdenes de los algoritmos.

Para su mejor comprensión, no dudes en ejecutar manualmente los algoritmos si es necesario.

3. Resolvé la siguiente recurrencia

$$t(n) = \begin{cases} 0 & \text{si } n = 0 \\ n^2 - n + t(n-1) & \text{si } n > 0 \end{cases}$$

4. Una balanza de dos platillos consiste de dos platillos y un mecanismo que se encarga de que los mismos se encuentren a igual altura solamente cuando ambos sostienen exactamente el mismo peso. En caso contrario, el platillo que sostiene mayor peso desciende y el otro se eleva. Para volver a equilibrar la balanza se utilizan pesitas de distintos pesos. Llamemos A y B a los platillos de la balanza y supongamos que tenemos 10 pesitas: p_1, p_2, p_3 y p_4 de 1 gramo cada una; p_5 de 5 gramos; p_6, p_7, p_8 y p_9 de 10 gramos cada una y p_{10} de 50 gramos. Si colocamos un objeto de 45 gramos en el platillo A, podemos equilibrar la balanza colocando las cinco pesitas p_5, p_6, p_7, p_8 y p_9 en platillo B. ¿Hay alguna manera de equilibrar la balanza con menos pesitas? Sí: podemos colocar la pesa p_{10} en el platillo B y la pesa p_5 en el platillo A junto al objeto en cuestión. Esta es la forma de equilibrar la balanza que menos pesitas requiere: dos.

   El problema de la balanza de dos platillos consiste en, dadas n pesitas de peso $p_1, \ldots, p_n$ y el peso P del objeto que se encuentra en el platillo A, determinar el menor número de pesitas necesarias para equilibrar la balanza. Las pesitas no necesitan estar ordenadas, sus pesos no necesariamente son los del ejemplo de arriba.

   Para resolver este problema es necesario utilizar backtracking, es decir, considerar todas las posibilidades. En este caso, cada pesita tiene tres posibilidades: ir al platillo A, ir al platillo B, y no ir a ninguno de los dos.

   Puede resultarte conveniente utilizar la siguiente definición para resolver el problema: $m(i,j)$ = "menor número de pesitas necesarias para equilibrar la balanza cuando el platillo A pesa j gramos más que el platillo B y se dispone solamente de las pesitas $1, 2, \ldots, i$." Da una definición recursiva de $m(i,j)$ que contemple todas las posibilidades, justificá claramente tu definición y cada uno de los casos considerados, y explicá cómo se utiliza la definición para calcular el menor número de pesitas necesarias para resolver el problema. Conviene permitir también que j sea negativo: significa que el platillo B pesa más que el A.

   En el ejemplo de arriba, $m(10,45) = 2$, $m(9,45) = 5$, $m(10,46) = 3$, $m(6,8) = 3$, $m(5,-4) = 2$, etc.

5. (Para alumnos libres) Explicá el peor caso del algoritmo de ordenación rápida, cuál es y por qué se vuelve ineficiente, y las distintas maneras de evitar ese problema. Mencioná otros algoritmos de ordenación eficientes que conozcas y comparalos con la ordenación rápica.