

Algoritmos y Estructuras de Datos II

Primer parcial – Miércoles 3 de mayo de 2017

Aclaraciones

- El parcial es a libro abierto.
- Cada ejercicio debe entregarse **en hojas separadas**.
- Incluir en cada hoja el número de orden asignado, número de hoja, apellido y nombre.
- Al entregar el parcial, completar el resto de las columnas en la planilla.
- Cada ejercicio se calificará con **P**romocionado, **A**probado, **R**egular, o **I**nsuficiente.
- El parcial estará aprobado si: (el ejercicio 1 tiene al menos **A**) y (el ejercicio 3 tiene al menos una **A** o (el ejercicio 3 tiene una **R** y el 2 al menos una **A**)).

Ej. 1. Especificación

El Centro de Gestión de Trámites (CGT) de la universidad cuenta con una intrincada red de ventanillas en las cuales se realizan distintos tipos de trámites.

En cualquier momento pueden llegar personas al CGT, las cuales saben exáctamente qué ventanillas tienen que visitar y en qué orden deben hacerlo, con lo cual se dirigen directamente a la primera de las ventanillas que debe visitar. Cuando una persona llega a una ventanilla, saca un número y espera a ser llamada. Una persona no visita más de una vez una misma ventanilla durante su estadía en el CGT.

Los empleados de las ventanillas no se toman tiempo libre, con lo cual cuando terminan de atender a una persona llaman inmediatamente a la que le sigue en la cola de espera. Si no hay nadie esperando, el empleado queda disponible para atender a la siguiente persona que llegue a la ventanilla. Por otro lado, cuando una persona termina de ser atendida en una ventanilla avanza hacia la siguiente ventanilla en su itinerario (a menos que no le queden trámites por hacer, en cuyo caso la persona se retira del establecimiento).

Modelar con un TAD el Centro de Gestión de Trámites descripto teniendo en cuenta que interesa saber:

- a) La composición actual de la cola en cada ventanilla, es decir, las personas que se encuentran esperando (o siendo atendidas) y en qué orden.
- b) La ventanilla que atendió a más personas.

Ej. 2. Complejidad

Discutir la veracidad de las siguientes afirmaciones, justificando adecuadamente en cada caso:

1. $\Omega(n) \subseteq O(n^2)$
2. $O(n^2) \subseteq \Omega(n)$
3. La complejidad temporal del *mejor caso* del Algoritmo 1 es $O(n^2)$.
4. La complejidad temporal del *peor caso* del Algoritmo 1 es $\Omega(n)$.

Algoritmo 1 Cuenta cuántos divisores tiene en el arreglo cada número par del mismo

```

1: function DIVISORESDEPARES(arreglo A)
2:   int i, total;
3:   total := 0;
4:   for i := 0 ... Long(A) - 1 do
5:     if 2 divide a A[i] then
6:       for j := 0 ... Long(A) - 1 do
7:         if A[j] divide a A[i] then
8:           total := total + 1;
9:         end if
10:      end for
11:    end if
12:  end for
13:  return total;
14: end function

```

Observación: Consideramos que las verificaciones de divisibilidad en el algoritmo son operaciones elementales.

Ej. 3. Diseño

El sitio REMATAZOS permite a sus usuarios publicar artículos para rematarlos al mejor postor. Mientras una publicación esté activa, los usuarios pueden ofertar un monto por el artículo (siempre mayor que el monto de la máxima oferta hasta el momento). Cada publicación cuenta con un *precio de venta automática* definido por el dueño de la misma (y oculto a la comunidad) y la duración de cada publicación es indefinida. Si en algún momento, una oferta alcanza o supera el precio de venta automática, el remate finaliza automáticamente concretándose así la venta. Por otro lado, el dueño del artículo puede decidir finalizar el remate en cualquier momento y en caso de haber alguna oferta, la venta se concreta al mejor postor. En cualquier caso, no se guardan registros de las publicaciones finalizadas. El siguiente TAD modela el sitio de remates descripto (aunque se omiten las axiomatizaciones).

TADs USUARIO y PUB son NAT

TAD OFERTA es TUPLA(USUARIO,NAT)

TAD REMATAZOS

observadores básicos

usuarios	:	rematazos	\rightarrow	conj(usuario)	
publicaciones	:	rematazos $r \times$ usuario u	\rightarrow	conj(pub)	$\{u \in \text{usuarios}(r)\}$
precioVentaAutom	:	rematazos $r \times$ pub p	\rightarrow	nat	$\{\text{publicacionActiva}(r,p)\}$
ofertas	:	rematazos $r \times$ pub p	\rightarrow	secu(oferta)	$\{\text{publicacionActiva}(r,p)\}$

generadores

iniciar	:		\rightarrow	rematazos	
altaUsuario	:	rematazos $r \times$ usuario u	\rightarrow	rematazos	$\{u \notin \text{usuarios}(r)\}$
publicar	:	rematazos $r \times$ usuario $u \times$ pub $p \times$ nat pr	\rightarrow	rematazos	$\{u \in \text{usuarios}(r) \wedge \neg \text{publicacionActiva}(r,p)\}$
ofertar	:	rematazos $r \times$ pub $p \times$ oferta of	\rightarrow	rematazos	$\{\Pi_1(of) \in \text{usuarios}(r) \setminus \{\text{dueño}(r,p)\} \wedge (\text{vacía?}(\text{ofertas}(r,p)) \vee_L \Pi_2(of) > \Pi_2(\text{ultimo}(\text{ofertas}(r,p))))\}$
finalizar	:	rematazos $r \times$ pub p	\rightarrow	rematazos	$\{\text{publicacionActiva}(r,p)\}$

otras operaciones

dueño	:	rematazos $r \times$ pub p	\rightarrow	usuario	$\{\text{publicacionActiva}(r,p)\}$
-------	---	------------------------------	---------------	---------	-------------------------------------

predicados

$\text{publicacionActiva}(r,p) \equiv (\exists u : \text{usuario}) (u \in \text{usuarios}(r) \wedge_L p \in \text{publicaciones}(r,u))$

Fin TAD

Se decidió utilizar la siguiente estructura para representar el TAD:

REMATAZOS **se representa con** estr

donde estr es tupla \langle usuarios: conj(usuario),
publicaciones: dicc(usuario,conj(tupla(pub,nat))),
dueños: dicc(pub, usuario),
ofertas: dicc(pub,secu(oferta)) \rangle

En esta estructura, *usuarios* almacena los usuarios del sitio, *publicaciones* registra las publicaciones activas de cada usuario (junto con sus precios de venta automática) y *dueños* indica qué usuario publicó cada artículo. Por otro lado, *ofertas* almacena todas las ofertas hechas (en orden) para cada publicación.

Teniendo en cuenta el TAD REMATAZOS y la estructura elegida para su representación se pide:

- Escribir en castellano el invariante de representación.
- Escribir formalmente el invariante de representación.
- Escribir formalmente la función de abstracción.