

Algoritmos y Estructuras de Datos I

Segundo cuatrimestre de 2016

26 de Agosto de 2016

TPE - OJOTA (Organización de Juegos Olímpicos Tp de Algoritmos 1) v1.0

Aclaraciones

- Para aprobar la totalidad del TP es necesario tener aprobado cada uno de sus módulos.
- **No** está permitido el uso de **acum** para la resolución.

Después de la destacada participación argentina en los Juegos Olímpicos de Río 2016, el Comité Olímpico Internacional (COI) decide instalar su oficina en nuestro país, y contratar a los alumnos de Algoritmos y Estructuras de Datos I para modelar ciertos problemas relacionados a los Juegos Olímpicos. Después de una serie de reuniones hemos podido recabar suficiente información acerca del problema a resolver, tomando las decisiones de diseño que se detallan a continuación. Nuestra misión, si decidimos aceptarla, es especificar todo lo que se pide a continuación.

1. Tipos

```

tipo Deporte = String;
tipo Pais = String;
tipo Sexo = Femenino, Masculino;

```

2. Atleta

```

tipo Atleta {
  observador nombre (a: Atleta) : String;
  observador sexo (a: Atleta) : Sexo;
  observador añoNacimiento (a: Atleta) :  $\mathbb{Z}$ ;
  observador nacionalidad (a: Atleta) : Pais;
  observador ciaNumber (a: Atleta) :  $\mathbb{Z}$ ;
  observador deportes (a: Atleta) : [Deporte];
  observador capacidad (a: Atleta, d: Deporte) :  $\mathbb{Z}$ ;
  requiere  $d \in \text{deportes}(a)$ ;

  invariante  $|\text{deportes}(a)| > 0$ ;
  invariante  $\text{sinRepetidos}(\text{deportes}(a))$ ;
  invariante  $\text{ordenada}(\text{deportes}(a))$ ;
  invariante  $\text{capacidadEnRango} : (\forall d \leftarrow \text{deportes}(a)) 0 \leq \text{capacidad}(a, d) \leq 100$ ;
}

```

- problema especialidad** (a: Atleta) = result : Deporte
Devuelve alguna de las especialidades del atleta a . Se consideran especialidades de un atleta aquellos deportes para los que mayor capacidad tiene.
- problema entrenarNuevoDeporte** (a: Atleta, d: Deporte, c: $\mathbb{Z}$)
Modifica al atleta a agregando al deporte d entre los deportes que practica. c representa la capacidad del atleta en ese deporte.

3. Competencia

```

tipo Competencia {
  observador categoria (c: Competencia) : (Deporte, Sexo);
  observador participantes (c: Competencia) : [Atleta];
  observador finalizada (c: Competencia) : Bool;
  observador ranking (c: Competencia) : [Atleta];
  requiere  $\text{finalizada}(c)$ ;
  observador lesTocoControlAntiDoping (c: Competencia) : [Atleta];
  requiere  $\text{finalizada}(c)$ ;
}

```

```

observador leDioPositivo (c: Competencia, a: Atleta) : Bool;
    requiere finalizada(c)  $\wedge$  a  $\in$  lesTocoControlAntiDoping(c);

invariante participaUnaSolaVez : sinRepetidos(ciaNumbers(participantes(c)));
invariante participantesPertenecenACat :
    ( $\forall p \leftarrow participantes(c)$ ) prm(categoria(c))  $\in$  deportes(p)  $\wedge$  sgd(categoria(c)) == sexo(p);
invariante elRankingEsDeParticipantesYNoHayRepetidos :
    finalizada(c)  $\Rightarrow$  incluida(ranking(c), participantes(c));
invariante seControlanParticipantesYNoHayRepetidos :
    finalizada(c)  $\Rightarrow$  incluida(lesTocoControlAntiDoping(c), participantes(c));
}

```

3. problema finalizarCompetencia (c: Competencia, posiciones: [Atleta], control: [(Atleta, Bool)])
Registra que una competencia terminó. La lista *posiciones* contiene a todos los participantes que finalizaron la competencia, en el orden en el que finalizaron. La lista *control* indica qué participantes debieron someterse al control antidopping, y cuál fue el resultado obtenido.
4. problema linfordChristie (c: Competencia, a: Atleta)
Modifica la competencia *c*, descalificando al atleta *a* de la competencia. Como este problema es para inconvenientes durante la competencia, la misma no debe estar finalizada.¹
5. problema gananLosMasCapaces (c: Competencia) = result : Bool
Indica si entre los participantes que finalizaron la competencia *c*, los que obtuvieron los mejores resultados eran los que más capacidad tenían para el deporte de dicha competencia.
6. problema sancionarTramposos (c: Competencia)
Modifica una competencia eliminando del ranking a aquellos participantes a quienes el control antidoping les dio positivo.

4. JJOO

```

tipo JJOO {
    observador año (j: JJOO) :  $\mathbb{Z}$ ;
    observador atletas (j: JJOO) : [Atleta];
    observador cantDias (j: JJOO) :  $\mathbb{Z}$ ;
    observador cronograma (j: JJOO, dia:  $\mathbb{Z}$ ) : [Competencia];
        requiere  $1 \leq dia \leq cantDias(j)$ ;
    observador jornadaActual (j: JJOO) :  $\mathbb{Z}$ ;

    invariante atletasUnicos : sinRepetidos(ciaNumbers(atletas(j)));
    invariante unaDeCadaCategoria : ( $\forall i, k \leftarrow [0..|competencias(j)|], i \neq k$ )
        categoria(competencias(j)i)  $\neq$  categoria(competencias(j)k);
    invariante competidoresInscriptos : ( $\forall c \leftarrow competencias(j)$ ) incluida(participantes(c), atletas(j));
    invariante jornadaValida :  $1 \leq jornadaActual(j) \leq cantDias(j)$ ;
    invariante finalizadasSiYaPasoElDia : lasPasadasFinalizaron(j)  $\wedge$  lasQueNoPasaronNoFinalizaron(j);
}

```

7. problema dePaseo (j: JJOO) = result : [Atleta]
Devuelve los atletas que fueron a los juegos pero no participan en ninguna competencia.
8. problema medallero (j: JJOO) = result : [(Pais, [$\mathbb{Z}$])]
Devuelve el medallero. Para cada país que haya ganado al menos una medalla, debe haber una lista con la cantidad de medallas de oro, plata, y bronce que ganó respectivamente. El medallero debe estar ordenado de acuerdo a la cantidad de medallas de oro, plata y bronce de cada país. Un país que no obtuvo medalla en ninguna de las categorías no debe aparecer.
9. problema boicotPorDisciplina (j: JJOO, cat: (Deporte, Sexo), p: Pais) = result : $\mathbb{Z}$
Modifica los JJOO retirando a todos los atletas del país *p* de la competencia de categoría *cat*, y devuelve la cantidad de atletas que se retiraron.
10. problema losMasFracasados (j: JJOO, p: Pais) = result : [Atleta]
De todos atletas nacidos en el país *p* que en más competencias participaron, devuelve aquellos que no ganaron ninguna medalla.

¹https://es.wikipedia.org/wiki/Linford_Christie

11. problema `liuSong` (j: JJOO, a: Atleta, p: País)
Modifica los juegos j , nacionalizando al atleta a al país p .²
12. problema `stevenBradbury` (j: JJOO) = `result` : Atleta
Entre todos los ganadores de una medalla de oro, devuelve al atleta con menos capacidad (considerando la capacidad que tiene en el deporte en el cual consiguió la medalla).³
13. problema `uyOrdenadoAsíHayUnPatrón` (j: JJOO) = `result` : Bool
Dado un Juego Olímpico, indica si al tomar la lista de los mejores países en cada día(*) hasta la jornada actual, los países que están en esa lista rotan de manera estricta.

(*) Por cada día, el *mejor país* es aquél que más medallas de oro ganó. En caso de haber un empate, nos quedaremos con el menor país comparando lexicográficamente. Si un día nadie ganó una medalla de oro, ese día se saltea y no aparece en la lista.

Ejemplo: La lista de los *mejores países* [Argentina, USA, Canada, Argentina, USA] rota estrictamente.
La lista de los *mejores países* [Argentina, USA, Canada, USA, Argentina] no rota estrictamente.
14. problema `sequíaOlímpica` (j: JJOO) = `result` : [País]
Dado un Juego Olímpico, devuelve todos los países participantes que más días pasaron sin ganar una medalla.
15. problema `transcurrirDia` (j: JJOO)
Modifica el juego reflejando que transcurrió un día. Los resultados de las competencias del día que pasó están dados por las capacidades de los atletas. En cada competencia se le realiza el control antidoping a un único atleta, elegido al azar, y el control le da positivo a lo sumo al 5 % de los atletas controlados.

5. Auxiliares

```

aux ciaNumbers (as: [Atleta]) : [Z] = [ciaNumber(a) | a ← as];
aux competencias (j: JJOO) : [Competencia] = [c | d ← [1..cantDias(j)], c ← cronograma(j, d)];
aux incluida (l1, l2: [T]) : Bool = (∀x ← l1) cuenta(x, l1) ≤ cuenta(x, l2);
aux lasPasadasFinalizaron (j: JJOO) : Bool = (∀d ← [1..jornadaActual(j)]) (∀c ← cronograma(j, d)) finalizada(c);
aux lasQueNoPasaronNoFinalizaron (j: JJOO) : Bool =
(∀d ← (jornadaActual(j)..cantDias(j))) (∀c ← cronograma(j, d)) ¬finalizada(c);
aux ordenada (l: [T]) : Bool = (∀i ← [0..|l| - 1]) li ≤ li+1;
aux sinRepetidos (l: [T]) : Bool = (∀i, j ← [0..|l|], i ≠ j) li ≠ lj;

```

²[https://es.wikipedia.org/wiki/Liu_Song_\(tenista_de_mesa\)](https://es.wikipedia.org/wiki/Liu_Song_(tenista_de_mesa))

³<https://www.youtube.com/watch?v=tYUjmEH9NNk>