Trabajo práctico 2: Diseño

Normativa

Límite de entrega: domingo 18 de octubre *hasta las 22:00 hs.* Enviar a algo2.dc@gmail.com
Normas de entrega: Ver "Información sobre la cursada" en el sitio Web de la materia.
(`http://www.dc.uba.ar/materias/aed2/2015/2c/informacion`)

Versión: 1.0 del 29 de septiembre de 2015 (ver TP2_Changelog.txt)

Enunciado

El objetivo de este TP es diseñar el módulo correspondiente al TAD CampusSeguro, junto con todos los módulos que sean necesarios para contar con un diseño **completo**.

Este enunciado contiene la especificación del TAD tal como deberá ser diseñado. Hay algunas diferencias en el comportamiento con respecto a lo pedido en el TP1:

- Los estudiantes y los hippies se identifican por nombre.
- Se agregaron dos otras operaciones al TAD principal:
  - conMismasSanciones: devuelve el conjunto de agentes que tienen la misma cantidad de sanciones que un agente dado.
  - conKsanciones: devuelve el conjunto de agentes que tienen exactamente k sanciones, sino devuelve vacío.

Estos cambios están especificados como son deseados. Sugerimos que confíen en la axiomatización más que en el castellano.

Contexto de uso y complejidades requeridas

Se requiere que las operaciones exportadas de los TADs tengan una contraparte en los módulos diseñados. Recomendamos modularizar adecuadamente: no necesariamente se aconseja hacer un único módulo por cada TAD.

Además, las operaciones indicadas a continuación deberán cumplir las complejidades temporales detalladas[1].

- campus(c) es $\mathcal{O}(1)$ y debe devolver un *campus* por referencia.
- estudiantes(c), hippies(c) y agentes(c) son $\mathcal{O}(1)$ y deben devolver un iterador.
- posEstudianteYHippie(n, c) es $\mathcal{O}(|n_m|)$, donde $|n_m|$ es la longitud más larga entre todos los nombres.
- posAgente(pl, c), cantSanciones(pl, c) y cantHippiesAtrapados(pl, c) son $\mathcal{O}(1)$ en el caso promedio. Asumir que el conjunto de números de placas de los ASs está uniformemente distribuido en el rango que cubre.
- ingresarEstudiante(n, p, c), ingresarHippie(n, p, c) y moverEstudiante(n, d, c) son $\mathcal{O}(|n_m|) + \mathcal{O}(\log N_a)$
- moverHippie(n, c) es $\mathcal{O}(|n_m|) + \mathcal{O}(\log N_a) + \mathcal{O}(N_e)$
- moverAgente(pl, c) es $\mathcal{O}(|n_m|) + \mathcal{O}(\log N_a) + \mathcal{O}(N_h)$
- másVigilante(c) y conMismasSanciones(a, c) son $\mathcal{O}(1)$
- conKSanciones(k, c) es $\mathcal{O}(N_a)$ la primera vez que se llama y $\mathcal{O}(\log N_a)$ en futuras llamadas mientras no ocurran sanciones.

donde:

- c es una instancia del campus
- p es una posición
- n es el nombre de un estudiante/hippie y $|n_m|$ es la longitud más larga entre todos los nombres del campus
- pl es el N° de placa de un agente

[1] Se desestimarán los costos de eliminación de elementos, con lo cual se pueden ignorar en el cálculo de complejidades.

- d es una dirección
- N_a es la cantidad de agentes
- N_e es la cantidad actual de estudiante
- N_h es la cantidad actual de hippies

Requisitos y consideraciones

- Todas las operaciones auxiliares deben ser especificadas formalmente según las herramientas vistas en clase. Agregar comentarios necesarios para entender la forma en la cual deben ser utilizadas para su correcto funcionamiento.
- Todos los algoritmos *deben* tener su desarrollo que justifique los órdenes de complejidad. Si algún paso es no trivial pueden hacer notas a continuación del mismo.
- Cuando se formalicen los invariantes y funciones de abstracción, *deben* identificar cada parte de la fórmula del Rep y comentar en castellano lo que describe.
- Tener en cuenta que las complejidades son en peor caso. Soluciones más eficientes serán bien recibidas.
- Tengan en cuenta que hay estructuras que pueden servir para más de una finalidad, sobre todos los contenedores.
- Pueden crear módulos adicionales si así lo necesitan.
- Cuentan con los siguiente TADs y módulos:
    - CHAR que representan los posibles caracteres. Siendo un tipo enumerado de 256 valores. con funciones ord y ord^{-1} para la correspondencia de cada $Char$ a Nat.
    - STRING como sinónimo de $Vector(Char)$.
    - Todos los definidos en el apunte de TADs básicos.
    - Todos los definidos en el apunte de módulos básicos.