DEPARTAMENTO DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo práctico 2: SistemaCNE

12 de noviembre de 2023 — Algoritmos y Estructuras de Datos

Integrante	LU	Correo electrónico
Romero Laino, Mauricio	18/23	mauricioromerolaino@gmail.com
Chiarizia, Luciano	757/22	chiarizialuciano@gmail.com
Manjarín, Santiago	616/22	santiagomanjarin111@gmail.com
Coronel, Facundo	445/23	coronelfacundo30@gmail.com

Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Ciudad Universitaria - (Pabellón I/Planta Baja)
Intendente Güiraldes 2610 - C1428EGA
Ciudad Autónoma de Buenos Aires - Rep. Argentina
Tel/Fax: (++54 +11) 4576-3300
http://www.exactas.uba.ar

Modulo NodoImpl **implementa** Nodo {
$var\ partido : int$
$var\ votosTotales : int$
$var\ votosCocientizados : int$
$var\ cociente : int$
}

Voy a describir el invariante del NodoImpl.

1. La variable partido es mayor o igual a 0.
2. El cociente es un número positivo.
3. La variable votosCocientizados es igual a la división entera entre votosTotales y cociente.
4. Las variables votosCocientizados y votosTotales son mayores o iguales a 0.

Modulo HeapEspecialImpl **implementa** HeapEspecial {
$var\ arr : Array < Nodo >$

}

Sea h': HeapEspecial. Voy a describir el invariante del HeapEspecialImpl.

1. arr cumple el siguiente predicado: esHeap(h'.arr)

   `pred esHeap` (arr: Array$< Nodo >$) {

   $(|arr| = 0) \vee_L (|arr| > 0 \wedge (\forall j : Z)(1 \leq j < |arr| \rightarrow_L arr[0].votosCocientizados \geq arr[j].votosCocientizados) \wedge (\forall i : Z)(0 \leq i < |arr| \rightarrow_L ((i < 2i + 1 < |arr| \rightarrow_L arr[i] \geq arr[2i + 1]) \wedge (i < 2i + 2 < |arr| \rightarrow_L arr[i] \geq arr[2i + 2]))))$

   }

Modulo SistemaCNEImpl **implementa** SistemaCNE {
$var\ partidos : Array < String >$
$var\ distritos : Array < String >$
$var\ ultimasMesasDistritos : Array < int >$
$var\ votosPresidenciales : Array < int >$
$var\ votosDiputados : Array < Array < int \gg$
$var\ votosTotalesDiputados : Array < int >$
$var\ primeroEnPresidencial : int$
$var\ segundoEnPresidencial : int$
$var\ votosTotalesPresidencial : int$
$var\ arrDeHeapDiputados : Array < HeapEspecial >$
$var\ diputadosPorDistrito : Array < int >$
$var\ memoBancasPorDistrito : Array < Array < int \gg$
$var\ flags : Array < bool >$
}

Sea s': SistemaCNE. Voy a ir describiendo el invariante de representación en palabras.

1. La variable partidos debe tener una longitud mayor o igual a 1, y su último elemento debe ser igual a "Blanco". $|s'.partidos| > 0 \ \wedge_L \ s'.partidos[|s'.partidos| - 1] = "Blanco"$
2. Las variables distritos, ultimasMesasDistritos, votosDiputados, arrDeHeapDiputados, diputadosPorDistrito, memoBancasPorDistrito, flags y votosTotalesDiputados deben tener la misma longitud.
3. Las variables partidos y votosPresidenciales deben tener la misma longitud. A su vez, todo elemento en memoBancasPorDistrito y arrDeHeapDiputados debe tener longitud $|s'.partidos| - 1$. Los elementos en votosDiputados deben tener longitud $|s'.partidos|$.
4. Las variables distritos y partidos no pueden tener nombres repetidos.
5. Las variables votosPresidenciales, diputadosPorDistrito, votosTotalesDiputados y ultimasMesasDistritos deben tener solo elementos no negativos.

6. La variable votosTotalesPresidencial es igual a la suma de todos los elementos en votosPresidenciales.

7. $|partidos| \geq 3$ implica luego que las variables primero y segundo deben moverse entre $0 \leq s'.primeroEnPresidencial, s'.segundoEnPresidencial < |s'.votosPresidencial| - 1$. Siempre van a ser desiguales: $s'.primeroEnPresidencial \neq s'.segundoEnPresidencial$ y además se cumple la propiedad de que primeroEnPresidencial es el partido (no blanco) con más votos en votosPresidenciales y segundoEnPresidencial es el partido (no blanco) con más votos en votosPresidenciales luego de primeroEnPresidencial.

8. La variable ultimasMesasDistritos debe cumplir que sus elementos se encuentran en un orden estrictamente creciente (no hay repetidos).

9. Cada elemento de votosDiputados contiene un arreglo de enteros, cada elemento que pertenezca a dicho arreglo debe ser un número no negativo.

10. El elemento en la posicion i-esima en s'.votosTotalesDiputados es igual a la sumatoria del arreglo en la posición i-ésima de s'.votosDiputados.

11. El elemento en la posición i-ésima en s'.flags es False si y solo si:
O bien, el heapEspecial de esa posición i-ésima del arrDeHeapsDiputados no está inicializado, o bien cumple lo siguiente:

    - Elemento en la posición i-ésima de s'.arrDeHeapDiputados es de tipo heapEspecial que todos sus nodos contiene en el atributo partido únicamente a todos los ids de los partidos, salvo el blanco, y no hay ids repetidos.
    - Además, cada nodo tiene los votosTotales asociados a los votos de diputados de un partido en dicho distrito. Es decir, se relaciona de esta manera con votosDiputados.
    - También el cociente tiene que valer 1 en ese caso.
    - La sumatoria del atributo votosTotales de cada nodo para ese distrito tiene que ser igual al valor de votosTotalesDiputados en dicho distrito.

12. Por otro lado, el elemento en la posición i-ésima en s'.flags es True si y solo si:

    - El elemento en la posición i-ésima en s'.memoBancasPorDistrito es un arreglo que contiene las bancas que le corresponden a cada uno de los partidos en dicho distrito según el sistema D'Hont (En particular, se relaciona con votosDiputados y con votosTotalesDiputados (ya que necesita superar el 3 % para ganar una banca)).
    - En este caso, la sumatoria del elemento en la posición i-ésima en s'.memoBancasPorDistrito es igual al elemento en la posición i-ésima en s'.diputadosPorDistrito. Es decir, estamos asignando la cantidad correctas de bancas de diputados en dicho distrito.