

Trabajo Práctico N° 2: MESSI +10

Lucas Eduardo Ramsés Perea, Matías Julián Tebele

Lunes 11 de Noviembre de 2013

Índice

1. Estructuras de datos implementadas	1
1.1. Lista Simplemente Enlazada	1
1.2. Pila Dinámica	2
1.3. Hash	2
1.4. ABB	2
1.5. Heap	2
1.6. Jugador	2
1.7. Equipo	2
1.8. Fixture	2
1.9. Sistema	3
2. Flujo del programa	3
2.1. Interacción con el usuario	3
2.2. Interpretación de las instrucciones recibidas por el usuario . .	3
2.3. Ejecución de las instrucciones	3
2.4. Fin de la ejecución	4
3. Resolución de los problemas planteados	4
3.1. Agregar resultado	4
3.2. Listar jugadores	4
3.3. Listar goleadores	4
3.4. Goles Jugador	5
3.5. Mostrar Resultado	5
4. Implementación	5
5. Aspectos a mejorar	5

1. Estructuras de datos implementadas

Hicimos uso de las siguientes estructuras previamente implementadas:

1.1. Lista Simplemente Enlazada

La lista se utiliza en la carga del fixture, para almacenar los nombres de los equipos participantes. También es empleada para almacenar listados de jugadores que son solicitados en el enunciado.

1.2. Pila Dinámica

Esta estructura es utilizada indirectamente en la implementación del TAD abb.

1.3. Hash

Usamos tablas de Hash para guardar los jugadores y los equipos. Cabe destacar que los elementos de la primera tabla son parte de los elementos almacenados en la segunda. De esta manera, la información se encuentra actualizada en ambas. Vimos necesario armar dos Hashes debido a que la especificación requiere que accedamos a jugadores y equipos en orden constante.

1.4. ABB

El árbol binario de búsqueda es utilizado para ordenar los jugadores de un equipo, cuando el usuario solicita un listado de los mismos organizado por nombre.

1.5. Heap

Usamos un heap para organizar a los jugadores bajo el criterio de goles convertidos.

1.6. Jugador

Es una estructura que engloba las características necesarias para identificar un jugador (equipo, nombre, dorsal) y su historial de goles.

1.7. Equipo

Agrupar jugadores, un nombre identificador y la cantidad de jugadores del plantel.

1.8. Fixture

Contiene los partidos a jugar en el torneo. Es el que determina las llaves y el desarrollo del torneo.

1.9. Sistema

Es la estructura encargada de administrar la información obtenida en el archivo. Contiene las estructuras de ordenamiento antes mencionadas, de manera tal que las consultas puedan ser efectuadas en los tiempos solicitados en la especificación.

2. Flujo del programa

El programa recibe un archivo de texto, el cual será procesado. Se creará un nuevo sistema, el cual obtendrá la información del archivo y la organizará en estructuras que manipularán las actualizaciones recibidas y facilitarán el acceso a la información existente. Una vez procesado todo el archivo, es cerrado y el sistema queda dispuesto para recibir consultas o actualizaciones.

2.1. Interacción con el usuario

El usuario enviará instrucciones al programa respetando el estándar especificado en el enunciado del TP. Si bien hay varias instancias de validación dentro del sistema, somos conscientes de que hay ciertos casos en los cuales el programa puede fallar debido a formatos inválidos. Esto podría ser un aspecto a mejorar.

2.2. Interpretación de las instrucciones recibidas por el usuario

Cada línea de texto (secuencia de caracteres finalizada con ENTER) será la instrucción a interpretar. Internamente, el programa almacena la línea en una cadena, que posteriormente es dividida en secciones. La primera sección es considerada el comando a realizar, mientras que el resto de la cadena representará el/los parámetros con los que operará el comando. De acuerdo a lo explicado en la subsección anterior, el programa validará el comando y posteriormente hará lo mismo con la relación comando - cantidad de parámetros.

2.3. Ejecución de las instrucciones

Una vez que los comandos y parámetros fueron validados correctamente, el programa procede a ejecutar la función correspondiente al comando ingresado. Esta es determinada dentro del programa principal, el cual administra la correspondencia entre comando y función a ejecutar y los parámetros a enviar a la función y es ejecutada dentro del módulo correspondiente al sistema.

de inscripciones. Las funciones del sistema, tras ser ejecutadas, devolverán valores que permitirán al programa principal imprimir la salida apropiada.

2.4. Fin de la ejecución

El programa finaliza al recibir en la entrada una señal de fin de archivo. Al finalizar, destruye el sistema, junto con sus estructuras asociadas.

3. Resolución de los problemas planteados

3.1. Agregar resultado

- Solución: procesa el identificador de ronda recibido y accede al partido asociado en el fixture. Consulta si el mismo puede jugarse y en caso afirmativo, agrega el resultado al partido y actualiza los goles de cada uno de los jugadores correspondientes a los equipos involucrados en el encuentro.
- Orden: El proceso se vale de operaciones constantes para acceder al partido y a los equipos. Para actualizar los goles, recorre los jugadores de cada plantel. Por lo tanto, el orden de la solución es $O(k)$, siendo k la cantidad de jugadores de los equipos participantes del partido.

3.2. Listar jugadores

- Solución: El sistema busca el equipo consultado y, en base al orden solicitado por el usuario, muestra la información de los jugadores de dicho equipo recorriendo el plantel.
- Orden: La obtención del equipo es constante y el listado de equipos recorre linealmente la estructura que almacena la información. Luego, el algoritmo resuelve el problema en $O(k)$, siendo k los jugadores a procesar.

3.3. Listar goleadores

- Solución: Se obtiene el máximo goleador del sistema y se imprime la información.
- Orden: $O(1)$, pues la obtención del goleador es constante, así como el relevamiento de los datos del mismo.

3.4. Goles Jugador

- Solución: El sistema accede al jugador solicitado e imprime información sobre el mismo.
- Orden: $O(1)$, pues la obtención del jugador y la devolución de la información correspondiente son constantes.

3.5. Mostrar Resultado

- Solución: Busca en el fixture el partido asociado al idr recibido y emite el informe del mismo.
- Orden: $O(1)$. Encuentro el partido y proceso la información en orden constante.

4. Implementación

5. Aspectos a mejorar

- Podría hacerse una validación más estricta respecto a lo que espera el sistema si consideramos las funciones implementadas.