Antonio Hernández, Ronald Bernal

Implementar un programa en C/C++ que realice la planeación y simulación de partidos de una liga mediante Round Robin recursivo.

Round Robin

Análisis y Diseño de Algoritmos

1. Análisis del problema
2. Descripción

Round-Robin es una técnica usada para seleccionar todos los elementos en un grupo de manera equitativa y en un orden racional, esto con el fin de lograr que cada elemento del grupo se junte únicamente una vez a cada uno de los demás elementos del grupo sin contarse a sí mismo.

Esta técnica se puede aplicar a la forma en cómo se distribuye un torneo, en donde cada equipo se enfrenta a todos los demás equipos en días distintos sin repetir equipos ni contarse a sí mismo.

Para resolver esto se obtiene el número de elementos/equipos a tratar, si el número n de elementos es par se debe implementar un algoritmo para lograr que cada elemento se enfrente a cada uno de los demás en n-1 movimientos/días, en caso de que n sea impar el número de movimientos/días será n.

1. Diseño de la solución
2. Algoritmo
3. Usar una estructura de datos donde los elementos/equipos del torneo se encontrarán guardados. En el caso de nuestro programa se utiliza un arreglo bidimensional para los enfrentamientos en donde en el eje X se representa el día y contra quién se enfrentará el equipo que se encuentra en el eje Y, además del arreglo para los partidos utilizamos otros 2 arreglos para guardar los resultados de los goles y las estadísticas de cada equipo.
4. Una vez que se tiene el número de elementos se crea la matriz con dimensiones [n][n-1] en caso de ser par o [n][n] en caso de ser impar.
5. Luego se ejecuta un algoritmo recursivo para los casos en los que n es una potencia de 2 o un algoritmo iterativo para los casos en los que n no es una potencia de 2.

|  |  |  |  |
| --- | --- | --- | --- |
|  | D1 | D2 | D3 |
| Eq1 |  | 3 |  |
| Eq2 |  | 4 |  |
| Eq3 |  | 2 |  |
| Eq4 |  | 1 |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | D1 | D2 | D3 |
| Eq1 | 2 | 3 |  |
| Eq2 | 1 | 4 |  |
| Eq3 |  | 2 |  |
| Eq4 |  | 1 |  |

1. En el algoritmo recursivo se obtiene el número de elementos, y en la posición [n/2+1][j] hasta que j = n/2 se le suma al número de equipo n/2, y a partir de j = n/2 +1 se le resta al número de equipo n/2. Para que quede así:

|  |  |  |  |
| --- | --- | --- | --- |
|  | D1 | D2 | D3 |
| Eq1 | 2 | 3 | 4 |
| Eq2 | 1 | 4 | 3 |
| Eq3 | 4 | 2 | 1 |
| Eq4 | 3 | 1 | 2 |

Y esto recursivamente hasta que queda así:

1. Una vez generados los partidos, se generan los resultados, esto se hace tomando en cuenta el peso de cada equipo, ya que se genera un número random en el rango del 0 al peso del equipo, esto permite que un equipo de peso 1 tenga la probabilidad de 1/10 de ganarle a un equipo de peso 0, obteniendo un marcador de 1-0 a favor del equipo de menor peso, sin embargo para evitar resultados exagerados de 10 – 0 implementamos una forma para dejar siempre la probabilidad de 1/10 y nivelar un poco más las cosas para cuando el score sea mayor que 5, esto se logra dividiendo el score entre 2 únicamente cuando este sea mayor que 5, de esta forma se conserva la probabilidad de 1/10 que un equipo de peso 1 le gane a uno de peso 10.
2. Implementación
3. Código C/C++

void generaPartidos(int x1, int y1, int sX, int sY){

if(sX != 0){

for(int i = 0; i < sY/2; i++){

matriz[x1+sX/2][y1+sY/2+i] = matriz[x1][y1+i];

}

int contY = 0;

for(int i = sY/2; i <sY; i++){

matriz[x1+sX/2][y1+contY] = matriz[x1][y1+sY/2+ contY];

contY++;

}

generaPartidos(x1, y1, sX/2, sY/2);

generaPartidos(x1 + sX/2, y1, sX/2, sY/2);

generaPartidos(x1, y1 + sY/2, sX/2, sY/2);

generaPartidos(x1 + sX/2, y1 + sY/2, sX/2, sY/2);

}

}

void calculaScores(){

srand(time(NULL));

for(int i = 0 ; i < N; i++){

for(int j = 1 ; j < N; j++){

tablaScores[i][j] = rand() % (peso.at(i)+1);

if(tablaScores[i][j]>5){

tablaScores[i][j] = tablaScores[i][j] /2;

}

}

}

}

void calculaEstadisticas(int eq1, int eq2, int goles1, int goles2){

//partidos jugados

tablaEstad[eq1][0]++;

tablaEstad[eq2][0]++;

//partidos ganados

if(goles1>goles2){

tablaEstad[eq1][1]++;

tablaEstad[eq2][3]++;

//pts

tablaEstad[eq1][7] += 3;

}

//partidos empatados

if(goles1==goles2){

tablaEstad[eq1][2]++;

tablaEstad[eq2][2]++;

//pts

tablaEstad[eq1][7] += 1;

tablaEstad[eq2][7] += 1;

}

//partidos perdidos

if(goles1<goles2){

tablaEstad[eq1][3]++;

tablaEstad[eq2][1]++;

//pts

tablaEstad[eq2][7] += 3;

}

//goles a favor

tablaEstad[eq1][4] +=goles1;

tablaEstad[eq2][4] +=goles2;

//goles en contra

tablaEstad[eq1][5] +=goles2;

tablaEstad[eq2][5] +=goles1;

//diferencia de goles

tablaEstad[eq1][6] = tablaEstad[eq1][4] - tablaEstad[eq1][5];

tablaEstad[eq2][6] = tablaEstad[eq2][4] - tablaEstad[eq2][5];

}

1. Documentación de las funciones

Las funciones principales del algoritmo son:

* void generaPartidos(int x1, int y1, int sX, int sY)
* void calculaScores()
* void calculaEstadisticas(int eq1, int eq2, int goles1, int goles2)

La función **generaPartidos** recibe diversos parámetros:

1. intx1: Es la posición en el eje X en la matriz donde se empezará a evaluar para calcular los partidos en los siguientes días.
2. intx1: Es la posición en el eje Y en la matriz donde se empezará a evaluar para calcular los partidos en los siguientes días.
3. intsX: Indica el tamaño en el eje X del cuadrante de la matriz que se está evaluando.
4. intsY: Indica el tamaño en el eje Y del cuadrante de la matriz que se está evaluando.

Ya que se está trabajando sobre la variable global int \*\*matriz, que contiene el número del equipo de referencia en la primera columna de la matriz que se va a enfrentar con los equipos en su misma fila pero en las columnas siguientes los respectivos días del torneo, lo que se hace es que se calcula recursivamente para todos los cuadrantes de la matriz en base al algoritmo de Round-Robin explicado anteriormente.

La función **calculaScores**.

Al igual que la función **generaPartidos** la función **calculaScores** trabaja sobre la variable global int \*\*tablaScores, que contiene el número de goles que metió el equipo con número relacionado a la posición en las filas en día relacionado a la posición en las columnas contra el equipo que jugaba ese día, el número de goles se calcula con un random del 0 al peso del equipo, de esta manera se mantienen las probabilidades que un equipo de mayor peso le gane a uno de menor peso, para lograr scores más reales y no tan exagerados de 10-0 decidimos nivelar que cuando un equipo de peso mayor que 5 logre un score mayor a 5 este se divida entre 2 para mantener scores reales y las probabilidades que un equipo de peso 1 solo tenga una única oportunidad de ganarle a un equipo de peso 2, además esto no afecta cuando se enfrentan equipos con ambos pesos mayores que 5, ya que si el score sin tratar fuera de 7-8 al “normalizarlo” sería 3-4 obteniendo al mismo ganador pero con un score un poco más real.

La función **calculaEstadisticas** recibe diversos parámetros:

1. int eq1: Es el equipo que se enfrentó con el eq2. El número del equipo que se relaciona con el respectivo número en la fila de la **tablaScores**, las columnas de la **tablaScores** son las estadísticas de este equipo.
2. int eq2: Es el equipo que se enfrentó con el eq1. El número del equipo que se relaciona con el respectivo número en la fila de la **tablaScores**, las columnas de la **tablaScores** son las estadísticas de este equipo.
3. int goles1: Es el número de goles que anotó el eq1 frente al eq2.
4. int goles2: Es el número de goles que anotó el eq1 frente al eq1.

Hace las comparaciones necesarias para determinar el número de partidos ganados, empatados y perdidos, así como los goles a favor, en contra y la diferencia de goles y el puntaje dentro del torneo.

1. Archivo de entrada

El archivo de entrada es de la siguiente forma:

4

Real\_Madrid 10

Valencia 9

Barcelona 10

Valladolid 7

Dónde en la primera línea debe ser un único elemento que será número que indique el número de equipos que van a jugar en el torneo. En las siguientes líneas deben haber únicamente 2 elementos, el primero el nombre del equipo y el segundo elemento el peso del equipo, opcionalmente la última línea estará dedicada a la fecha de inicio del torneo, esto para simularlo de una forma más real.

1. Pruebas
2. Prueba 1

Archivo de entrada:

8

Real\_Madrid 10

Valencia 9

Barcelona 10

Valladolid 7

Machester 5

Atletico\_Madrid 4

Bordeux 3

Juventus 1

Resultado:

Torneo Round Robin

1- Imprime Calendario de Juegos

2- Simular liga completa

3- Simular siguiente por fechas

4- Imprime Estadisticas

5- Salir de la Liga

2

Round Robin

Jornada 1 del torneo

Real\_Madrid VS Valencia

4-4

Barcelona VS Valladolid

3-0

Machester VS Atletico\_Madrid

0-1

Bordeux VS Juventus

2-0

Jornada 2 del torneo

Real\_Madrid VS Barcelona

5-5

Valencia VS Valladolid

5-3

Machester VS Bordeux

1-0

Atletico\_Madrid VS Juventus

4-1

Jornada 3 del torneo

Real\_Madrid VS Valladolid

3-3

Valencia VS Barcelona

5-3

Machester VS Juventus

0-0

Atletico\_Madrid VS Bordeux

2-3

Jornada 4 del torneo

Real\_Madrid VS Machester

5-3

Valencia VS Atletico\_Madrid

3-1

Barcelona VS Bordeux

0-2

Valladolid VS Juventus

3-0

Jornada 5 del torneo

Real\_Madrid VS Atletico\_Madrid

4-3

Valencia VS Machester

2-0

Barcelona VS Juventus

3-0

Valladolid VS Bordeux

3-3

Jornada 6 del torneo

Real\_Madrid VS Bordeux

0-1

Valencia VS Juventus

2-0

Barcelona VS Machester

3-5

Valladolid VS Atletico\_Madrid

2-0

Jornada 7 del torneo

Real\_Madrid VS Juventus

3-1

Valencia VS Bordeux

3-3

Barcelona VS Atletico\_Madrid

4-4

Valladolid VS Machester

3-5

Tabla Estadisticas

PJ PG PE PP GF GC DG PTS

7 3 3 1 24 20 4 12 Real\_Madrid

7 5 2 0 24 14 10 17 Valencia

7 2 2 3 21 21 0 8 Barcelona

7 2 2 3 17 19 -2 8 Valladolid

7 3 1 3 14 14 0 10 Machester

7 2 1 4 15 17 -2 7 Atletico\_Madrid

7 4 2 1 14 9 5 14 Bordeux

7 0 1 6 2 17 -15 1 Juventus

1. Prueba 2

Archivo de entrada:

4

Real\_Madrid 10

Valencia 9

Barcelona 10

Valladolid 7

Resultado:

Torneo Round Robin

1- Imprime Calendario de Juegos

2- Simular liga completa

3- Simular siguiente por fechas

4- Imprime Estadisticas

5- Salir de la Liga

2

Round Robin

Jornada 1 del torneo

Real\_Madrid VS Valencia

3-4

Barcelona VS Valladolid

3-3

Jornada 2 del torneo

Real\_Madrid VS Barcelona

3-0

Valencia VS Valladolid

4-2

Jornada 3 del torneo

Real\_Madrid VS Valladolid

3-3

Valencia VS Barcelona

3-1

Tabla Estadisticas

PJ PG PE PP GF GC DG PTS

3 1 1 1 9 7 2 4 Real\_Madrid

3 3 0 0 11 6 5 9 Valencia

3 0 1 2 4 9 -5 1 Barcelona

3 0 2 1 8 10 -2 2 Valladolid

1. Reflexiones grupales

Con este proyecto el grupo aprendió que la combinación de las técnicas de algoritmos tales como divide y vencerás pueden resolver problemas cotidianos tales como un torneo tipo Round-Robin de una forma eficiente.