Programación II

Facultad de Estudios Estadísticos
Universidad Complutense de Madrid
Curso 2020-2021

"El juego es la forma más elevada de la investigación" (Albert Einstein)

Primera práctica obligatoria. El juego del Nim

En el **Nim**, los jugadores colocan una cantidad arbitraria de fichas sobre la superficie del juego, separadas en grupos. Tanto la cantidad de grupos como la cantidad de fichas en cada grupo son también arbitrarias. En cada turno, el jugador correspondiente quita cualquier cantidad de fichas de un solo grupo (entre uno y la cantidad de fichas del grupo), y los jugadores se alternan retirando fichas. La partida finaliza cuando un jugador quita la última ficha de la superficie del juego, declarándose entonces ganador.

El propósito de la práctica es codificar algunas funciones del juego. Para ello, asumimos que la cantidad de grupos es diez y la cantidad máxima de fichas en cada grupo es 20. El estado de una partida se puede representar mediante un vector de enteros de tamaño diez como el siguiente:

3	7	3	15	12	0	9	0	1	6

1. Diseña una función llamada `initGame()` que prepare una partida, insertando los valores introducidos por el teclado en un vector de enteros de tamaño adecuado. La función debe comprobar que los datos insertados en el vector son válidos.
2. Diseña una función llamada `deletePieces()` que, dados el vector v de una partida, una posición p y una cantidad de fichas c, elimine la cantidad de fichas c de la posición p del vector v. La función debe comprobar que tanto la posición p como la cantidad de fichas c son válidas.
3. Diseña una función llamada `endGame()` que decida si una partida dada ha terminado, es decir, si su vector v asociado contiene únicamente valores nulos.
4. En ciertas ocasiones, es posible emplear estrategias concretas como la **estrategia Denier**:

   a) Diseña una función llamada `isDenierForm()` que determine si todos los grupos de una partida dada tienen una sola ficha excepto uno, que tiene varias piezas. En caso afirmativo, la función devolverá el grupo que tiene tal cantidad de fichas y en caso contrario, la función devolverá -1. Por ejemplo, la partida siguiente es una partida Denier y la función `isDenierForm()` devolverá el valor 3.

1	1	1	12	1	1	1	1	1	1

   b) Diseña una función llamada `maxPiecesGroup()` que determine el grupo de una partida dada que tiene la mayor cantidad de fichas. Si la partida tiene varios grupos con tal cantidad, entonces la función devolverá el primero de ellos.

   c) Diseña una función llamada `deletePiecesDenier()` que aplique la estrategia siguiente:

   - Si la partida es Denier y la cantidad de fichas del grupo con varias piezas es impar, entonces la función eliminará la mitad de la fichas de ese montón. Si la partida es Denier y la cantidad de fichas del grupo con varias piezas es par, entonces la función eliminará la mitad más uno de la fichas de ese montón. Por ejemplo, en la partida Denier anterior, la función eliminará siete fichas del grupo en la posición tres.
   - Si la partida no es Denier, entonces la función eliminará una ficha del grupo con la mayor cantidad de fichas.

5. Diseña una función llamada `playGame()` que prepare una partida y efectúe una cantidad de jugadas dada empleando la estrategia Denier. La partida debe finalizar tan pronto como se hayan eliminado todas las fichas de todos los grupos o se haya alcanzado la cantidad de jugadas dada.