# Introducción a la Programación – Algoritmos y Programación
# UNQ – 2do semestre de 2013 – Primer parcial – Nurikabe II

Nurikabe es un juego en el que un jugador debe dibujar un río en un mapa de forma tal de generar un conjunto de islas que satisfagan ciertas particularidades. El *mapa* es una grilla rectangular formada por celdas. Inicialmente algunas de estas celdas contienen un número y el resto se encuentran vacías (ver Figura 1(a)). Las celdas con número se llaman *indicadoras*. A medida que el juego transcurre, el jugador va *pintando* algunas de las celdas vacías con un lápiz negro; estas celdas negras van formando los segmentos del río, y dan origen a la formación de islas. Una *isla* es simplemente una región sin celdas negras[1] que está bordeada por el río o por los bordes del mapa (ver Figura 1(b)). El objetivo del juego es dibujar el río de forma tal que al finalizar se satisfagan las cuatro condiciones siguientes (ver Figuras 1(c) y 1(d)).

- Cada isla tiene una única celda indicadora (el resto de sus celdas están vacías).
- Cada isla tiene tantas celdas como el número que indica su única celda indicadora.
- El río es continuo, es decir, se debe poder recorrer todas las celdas negras con movimientos horizontales y verticales sin pasar por celdas blancas.
- No hay ninguna región de $2 \times 2$ celdas que contenga sólo celdas negras.

El objetivo del presente parcial es programar parte del juego Nurikabe. Para representar el mapa vamos a utilizar el tablero. Cada celda pintada se representa poniendo una bolita negra, las celdas indicadoras tienen tantas bolitas azules como el número correspondiente, y las celdas vacías no tienen bolitas (ver Figura 2).

Para simplificar la tarea de programar el juego, nos conviene tener un procedimiento que nos permita marcar cada segmento del río, usando bolitas **rojas**. Decimos que una celda está *marcada* si contiene una bolita roja y *desmarcada* en caso contrario.

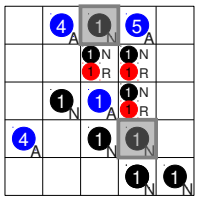
### EJERCICIO 1 (2 puntos)

*Una celda es* marcable *cuando (a) es parte del río (b) está desmarcada, y (c) es lindante con alguna celda marcada. Escribir una función* `hayCeldaMarcable` *que indique si el tablero tiene alguna celda marcable.*

***Sugerencia:*** *escribir una función* `esCeldaMarcable` *que determine si la celda actual es marcable.*

***Usar sin definir:*** `lindanteMarcada(d)` *que denota* `True` *cuando existe una celda en dirección* `d` *que además está marcada*[2]

***Ejemplo:*** *las celdas resaltadas son las* ***únicas*** *celdas marcables, porque están desmarcadas, forman parte del río y son lindantes (al N, E, S u O) a celdas marcadas.*



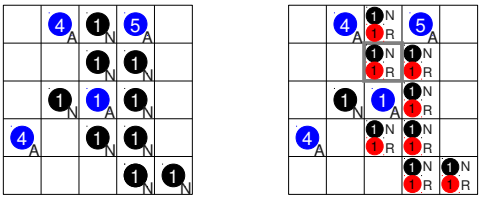
### EJERCICIO 2 (1 punto)

*Escribir un procedimiento* `MarcarSegmentoRio` *que marque todas las celdas del segmento del río que contiene la celda actual. Para ello,* ***debe*** *comenzar marcando la celda actual, y luego* ***debe*** *marcar todas las celdas marcables hasta que no queden más celdas marcables.*[3]

***Precondición:*** *no hay ninguna celda marcada en el tablero antes de la invocación del procedimiento. La celda actual forma parte del río.*

***Usar sin definir:*** `IrACeldaMarcable` *que posiciona el cabezal en una celda marcable. Como precondición, debe haber al menos una celda marcable en el tablero.*

***Ejemplos:*** *El tablero de la derecha resulta de aplicar* `MarcarRio` *en el tablero de la izquierda cuando el cabezal se encuentra en la celda recuadrada.*



Recordemos que para resolver un mapa hay que dibujar el río de forma tal que cada isla contenga exactamente una celda indicadora, de forma tal que el río sea continuo y no tenga lagos. Las siguientes funciones se usan para verificar las propiedades del río.

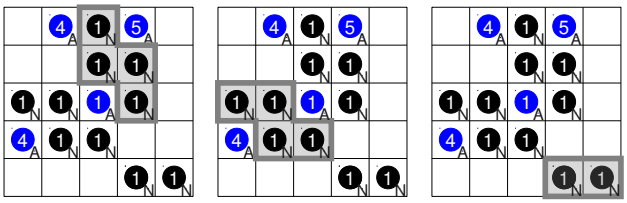
### EJERCICIO 3 (0.5 puntos)

*Escribir la función* `tamañoSegmentoRio` *que denote la cantidad de celdas del segmento del río que contiene la celda actual.*

***Precondición:*** *no hay ninguna celda marcada en el tablero antes de invocar la función. La celda actual forma parte del río.*

***Observación:*** *recordar la función* `nroBolitasTotal(c)` *(ejercicio 16, práctica 5).*

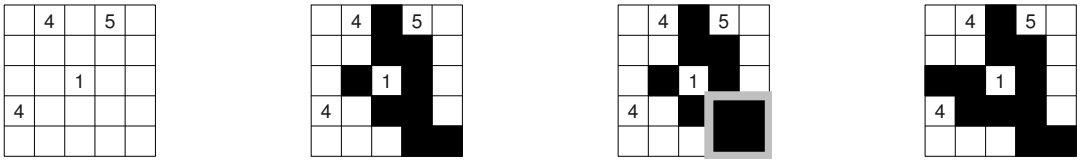
***Ejemplos:*** *de izquierda a derecha, los tamaños de los segmentos del río son 4, 4 y 2.*



[1] Es decir, contiene celdas blancas o celdas indicadoras.

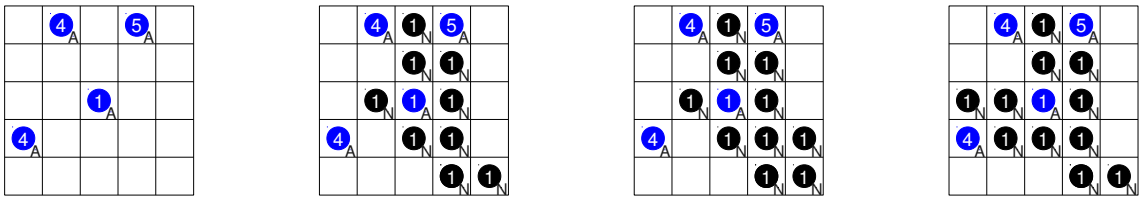
[2] Notar que `lindanteMarcada` no tiene precondición. En caso que la celda actual sea la última en dirección `d`, la función denota `False`.

[3] La estrategia propuesta funciona. No hace falta justificar este hecho, basta con implementar lo que se indica.



(a) (b) (c) (d)

Figura 1: (a) Mapa inicial con cuatro casillas indicadoras. (b) Resolución parcial del mapa inicial con tres islas y un río que no es continuo; notar, además, que una de las islas contiene dos indicadores. (c) Solución del mapa inicial que es incorrecta porque tiene una región de $2 \times 2$ celdas negras (remarcadas con gris), y porque la isla que contiene el indicador 5 tiene sólo 4 celdas. (d) Solución del mapa inicial que es correcta porque el río es continuo, cada isla tiene la misma cantidad de celdas que la que indica su única casilla indicadora, y no hay regiones de $2 \times 2$ celdas negras.



(a) (b) (c) (d)

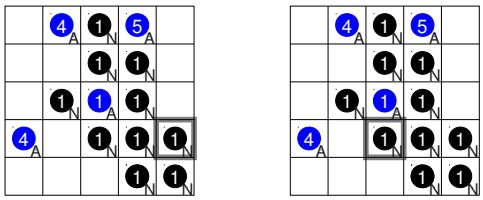
Figura 2: Representaciones de los mapas de la Figura 1 en Gobstones.

## EJERCICIO 4 (3 puntos)

*Un* lago *es una región de* $2 \times 2$ *celdas del río (ver Figura 1 (c)). Escribir una función* `estaEnLago` *que indique si la celda actual es parte de un lago.*

***Usar sin definir:*** `hayRioAl(d)` *que denota* `True` *si la celda en dirección* `d` *es parte del río; como precondición, debe ser posible moverse en dirección* `d`*.* `hayRioDiag(d)` *que denota* `True` *si la celda en dirección* `d + siguiente(d)` *es parte del río; como precondición, debe ser posible moverse tanto en dirección* `d` *como* `siguiente(d)`*.*

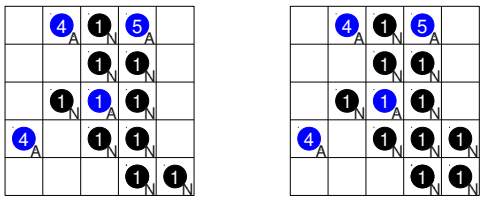
***Ejemplo:*** *en el tablero de la izquierda la celda bajo el cabezal —recuadrada en gris— forma parte de un lago, mientras que en el tablero de la derecha la celda bajo el cabezal no forma parte de un lago.*



## EJERCICIO 5 (1 punto)

*Escribir la función* `nroCeldasEnLago` *que denota la cantidad de celdas del río que pertenecen a un lago.*

***Ejemplo:*** *En el tablero de la izquierda hay 0 celdas en un lago, mientras que en el de la derecha hay 4.*



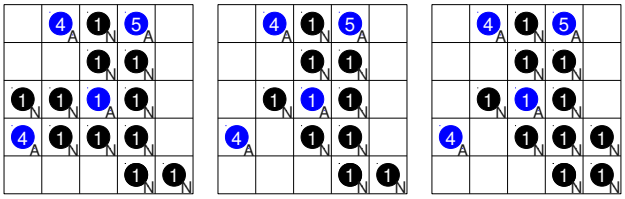
## EJERCICIO 6 (0.5 puntos)

*Escribir la función* `rioCorrecto` *que denota* `True` *si (a) el río es continuo y (b) no contiene lagos. En caso en que no haya celdas pintadas, la función debe retornar* `True`*.*

***Precondición:*** *no hay ninguna celda marcada en el tablero antes de invocar la función.*

***Usar sin definir:*** `IrARio` *que posiciona el cabezal en alguna celda del río; como precondición, debe haber al menos una celda del río.* `nroCeldasRio` *que denota la cantidad* ***total*** *de celdas que forman parte del rio.*

***Ejemplos:*** *en el tablero de la izquierda el río es correcto, el del centro no es correcto porque no es continuo y el de la derecha no es correcto porque tiene un lago.*



## EJERCICIO 7 (2 puntos)

*Decimos que una celda es* conectora *cuando (a) no esta pintada, (b) no es indicadora y (c) el río es correcto luego de pintarla (no importa si antes era correcto o no). Escribir una función* `quedanCeldasConectoras` *que denota* `True` *si hay al menos una celda conectora en el tablero.*

***Precondición:*** *no hay ninguna celda marcada en el tablero antes de invocar la función.*

***Ejemplos:*** *En el tablero de la izquierda las celdas marcadas son conectoras, mientras que el tablero de la derecha no contiene celdas conectoras.*