

Introducción a la Programación – UNQ – 2^{do} semestre de 2013

Primer parcial – NURIKABE I

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 negras¹ 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 2×2 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 isla del mapa, 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 de una isla (b) está *desmarcada*, y (c) es *lindante* con alguna celda marcada. Escribir una función `hayCeldaMarcable` que indique si el tablero tiene alguna celda marcada.

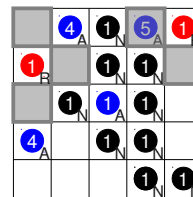
Sugerencia: escribir una función `esCeldaMarcable` que determine si la celda actual es *marcable*.

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* marcables, porque están *desmarcadas*, forman parte de una isla y son *lindantes* (al N, E, S u O) a celdas marcadas.

¹Es decir, contiene celdas blancas o celdas indicadoras.

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



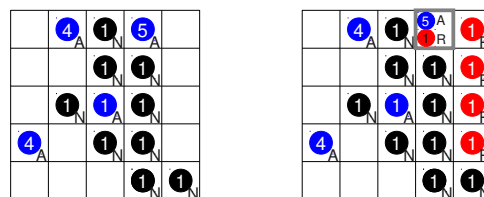
EJERCICIO 2 (1 punto)

Escribir un procedimiento `MarcarIsla` que marque todas las celdas de la isla 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 de una isla.

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

Ejemplos: El tablero de la derecha resulta de aplicar `MarcarIsla` 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. Más aún, cada isla debe tener tantas celdas como el número que aparece en la celda indicadora (ver Figura 1 (d)). Las siguientes funciones se utilizan para verificar qué islas fueron resueltas.

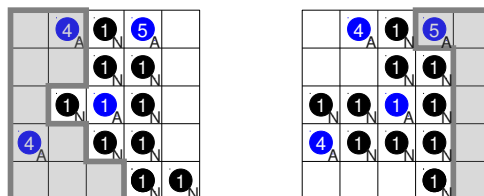
EJERCICIO 3 (0.5 puntos)

Escribir la función `tamañoIsla` que denote la cantidad de celdas de la isla que contiene la celda actual.

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

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

Ejemplos: el tamaño de la isla de la izquierda es 10, el de la derecha es 6.



EJERCICIO 4 (1 punto)

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

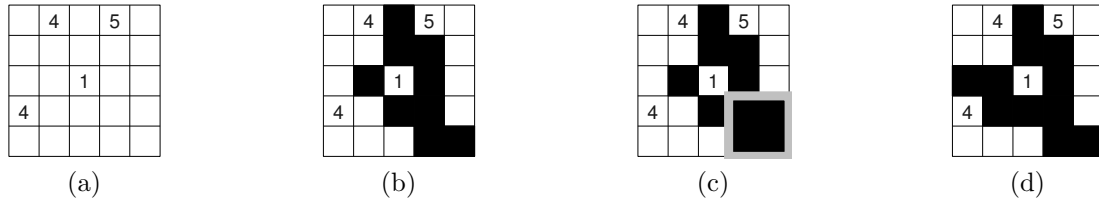


Figura 1: (a) Mapa inicial con cuatro casillas indicadoras. (b) Resolución parcial del mapa inicial con tres islas y un río que no es continuo; notar, además, que una de las islas contiene dos indicadores. (c) Solución del mapa inicial que es incorrecta porque tiene una región de 2×2 celdas negras (remarcadas 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 2×2 celdas negras.

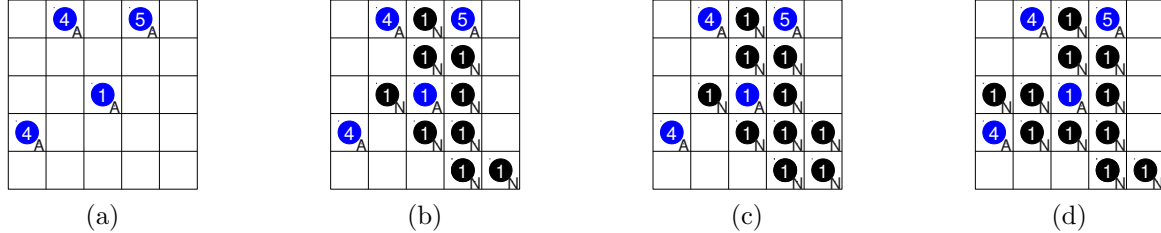
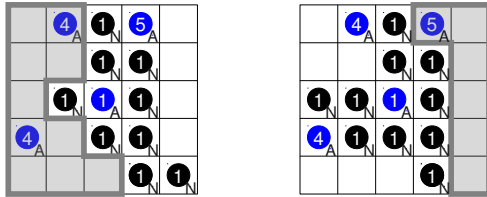


Figura 2: Representaciones de los mapas de la Figura 1 en Gobstones.

Escribir la función `nroIndicadoras` que denota la cantidad de celdas indicadoras de la isla que contiene la celda actual.

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

Ejemplo: la isla de la izquierda tiene 2 indicadores, el de la derecha 1.



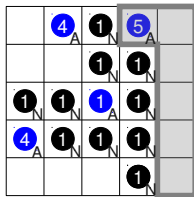
EJERCICIO 5 (0.5 puntos)

Escribir la función `indicador` que denote el valor del indicador de la isla que contiene la celda actual.

Precondición: hay una única celda indicadora en la isla. No hay ninguna celda marcada antes de invocar la función. La celda actual forma parte de una isla.

Usar sin definir: `IrAIndicadorMarcado` que posiciona el cabezal en alguna celda indicadora que está marcada. Como **precondición**, tiene que haber al menos una celda indicadora marcada.

Ejemplo: el indicador de la isla marcada es 5.



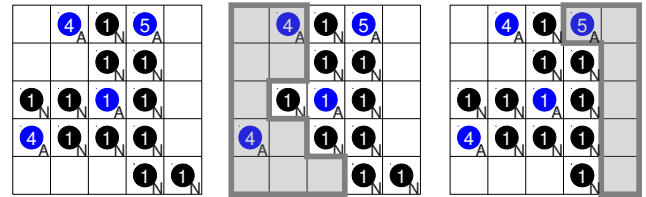
EJERCICIO 6 (2 puntos)

Decimos que una isla está resuelta si (a) contiene exactamente una celda indicadora, y (b) la cantidad de celdas de la isla es igual al número de la celda indicadora. Escribir la función `quedanIslasSinResolver` que retorna `true` si el mapa contiene alguna isla no resuelta.

Precondición: no hay ninguna celda marcada en el table-

ro antes de invocar la función.

Ejemplos: en el tablero de la izquierda todas las islas están resueltas; en el tablero del centro la isla marcada no está resuelta dado que contiene dos celdas indicadoras; en el tablero de la derecha la isla marcada no está resuelta porque tiene 6 celdas y su indicador es 5.



Para finalizar, la siguiente función se usará para determinar si el río no tiene regiones pintadas de 2×2 celdas.

EJERCICIO 7 (3 puntos)

Un lago es una región de 2×2 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.

