

Preparación para el Primer Parcial

- Si bien el presente no es un examen, puede ser beneficioso intentar realizarlo dentro de un marco de tiempo equivalente al que tendrá en el examen, en un ambiente sin interrupciones y de forma individual en papel.
- También recomendamos aspirar a escribir de forma prolija, o pasar el contenido de hojas borradores a hojas finales para poder estimar correctamente durante el los tiempos que esta tarea demanda durante el examen.
- Los procedimientos procedimientos y funciones primitivos que son dados en el examen deberán ser implementados en caso de que se desee probar la solución en máquina, por lo que recomendamos que aspire a realizar esto solamente una vez terminado de realizar los ejercicios en papel.
- Aconsejamos leer el enunciado completo antes de empezar a resolver los ejercicios, ya que ayuda a comprender mejor el dominio.
- Recuerde que el parcial será a libro abierto, que puede consultar cualquier material (en papel) que haya sido escrito antes de comenzar el examen y usar sin definir todas las funciones y procedimientos vistos durante la cursada. Tenga esto en cuenta al desarrollar su solución.
- Pensar bien la estrategia a seguir, expresando la misma mediante procedimientos y funciones. Recordar escribir primero los contratos.

Convención de Motoqueros

La Convención de Motoqueros es un juego que consiste en simular el comportamiento de motoqueros en una convención. Un motoquero puede estar activo o inactivo. A los motoqueros les gusta interactuar con otros motoqueros. Como producto de esta interacción algunos motoqueros pasan a la inactividad (duermen) y otros se hacen activos (despiertan).

La interacción se produce entre motoqueros activos que se mueven cerca unos de otros (decimos que son vecinos en ese caso). La interacción entre motoqueros activos sigue las siguientes pautas (de comportamiento):

- a. Todo motoquero activo con menos de dos vecinos activos se inactiva (no puede lucirse).
- b. Todo motoquero activo con más de tres vecinos activos pasa a la inactividad (le es imposible destacarse).
- c. Todo motoquero activo con exactamente dos o tres vecinos activos continúa activo.
- d. Todo motoquero inactivo con exactamente tres vecinos activos se convierte en activo (el grupo lo incentiva).

Estas pautas se aplican simultáneamente sobre todas los motoqueros de la convención. Cada una de estas aplicaciones simultáneas de las pautas se

conoce como una fase. Cada fase produce un nuevo estado de la convención.

La Convención de Motoqueros se puede modelar en Gobstones. Cada motoquero puede representarse con una celda del tablero. La misma tiene una bolita verde si el motoquero que representa está activo y ninguna verde si está inactivo.

Una motoquero es vecino de otro si sus correspondientes celdas son lindantes al N, NE, E, SE, S, SO, O u NO. La figura 1 muestra una convención posible y exhibe cómo evoluciona en una fase.

Además, durante la convención, se arman clubes. Es decir, grupos de motoqueros activos que están conectados entre sí. Dos motoqueros se dicen conectados cuando hay una secuencia de motoqueros activos vecinos que los unen.

Podemos además asignarles números a cada club, para diferenciarlos unos de otros y determinar cosas como qué grupo tiene más miembros, o cuál tiene menos.

La figura 2 muestra diversos grupos en una convención, habiendo asignado un número a cada grupo.

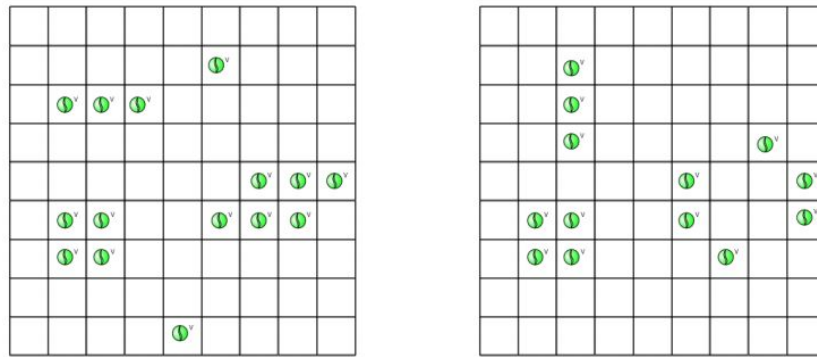


Figura 1: A la izquierda se ve un posible estado inicial de una convención de motoqueros. A la derecha se ve la misma convención luego de la primer fase.

Para poder procesar la convención de motoqueros en Gobstones vamos a requerir de una serie de procedimientos y funciones primitivas que se enuncian a continuación:

fueIncentivado()

PROPÓSITO: Indica Verdadero si el motoquero de la celda actual fué incentivado en esta fase a partir de interactuar con otros motoqueros. Es decir, en la próxima fase pasará a estar activo.

PRECONDICIÓN: Ninguna.

seAburrió(palo)

PROPÓSITO: Indica Verdadero cuando el motoquero actual se aburrió en esta fase, ya sea porque no pudo lucirse o porque le fue imposible destacarse, y por tanto pasará a estar inactivo en la próxima fase.

PRECONDICIÓN: Ninguna.

IncentivarAlMotoquero()

PROPÓSITO: Incentiva al motoquero actual, de forma que en la próxima fase deberá estar activo.

PRECONDICIÓN: El motoquero actual no fue incentivado previamente en esta fase ni tampoco se aburrió.

AburrirAlMotoquero()

PROPÓSITO: Aburre al motoquero actual, de forma que en la próxima fase deberá estar inactivo.

PRECONDICIÓN: El motoquero actual no fue aburrido previamente en esta fase ni fue incentivado.

ActualizarLaconvención()

PROPÓSITO: Actualiza la convención de motoqueros, pasando a una próxima fase. Para ello, este procedimiento transforma a todos los motoqueros aburridos en motoqueros inactivos, a todos los motoqueros incentivados en motoqueros activos, y deja igual a aquellos motoqueros que no están ni aburridos ni incentivados. Al terminar, el humor de los motoqueros de la convención es neutro (es decir, no hay ningún motoquero incentivado o aburrido en la convención).

PRECONDICIÓN: Ninguna.

MarcarMotoqueroComoPertenecienteAlClub_(número)

PROPÓSITO: Marca al motoquero actual como perteneciente al club número "número".

PRECONDICIÓN: El motoquero actual se encuentra activo y no pertenece todavía a ningún club.

elMotoqueroPerteneceAUnClub()

PROPÓSITO: Describe Verdadero si el motoquero actual pertenece a algún club. Falso en caso contrario.

PRECONDICIÓN: El motoquero actual se encuentra activo.

clubAlQuePerteneceUnMotoquero()

PROPÓSITO: Describe el número al cual pertenece el motoquero actual.

PRECONDICIÓN: El motoquero actual se encuentra activo y pertenece a un club.

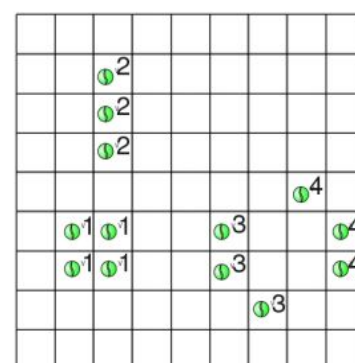


Figura 3. Hay 4 clubes distintos en la convención, siendo el club número 1 aquel que tiene más miembros (4 miembros)

EJERCICIOS

Ejercicio 1)

Escribir la función `elMotoqueroEstáActivo()` que retorna Verdadero si el motoquero sobre el cuál se encuentra el cabezal está activo y Falso en caso de que esté inactivo.

Ejercicio 2)

Escribir la función `cantidadDeVecinosActivos()` que describe la cantidad de vecinos activos que tiene el motoquero sobre cuál se encuentra el cabezal.

Ejercicio 3)

Escribir el procedimiento `ProcesarMotoquero()` que aplique las pautas de la fase al motoquero actual. Note que no se pueden aplicar las pautas de la fase de forma directa sobre un único motoquero, pues inactivarlo o activarlo en esta fase haría que, al procesar el motoquero vecino, este tenga menos o más vecinos activos que los que debería. Por esto, debemos simplemente aburrir al motoquero (si se va a inactivar en la próxima fase) o incentivarlo (si se va a activar la próxima fase).

Ejercicio 4)

Escribir el procedimiento `Simular_Fases(cantidadDeFases)` que dado un número positivo `cantidadDeFases` simule la interacción de la convención de motoqueros durante tantas fases como indique dicho número.

Ejercicio 5)

Se desea realizar el procedimiento `cantidadDeClubesEnLaConvención()` que determine la cantidad total de clubes de motoqueros que hay en la convención. Para ello se ha planteado una estrategia top-down que propone que para contar los clubes, lo más fácil es asignar un número de club distinto y consecutivo a cada club, hasta no tener más clubes. El último número de club asignado debería permitir determinar la cantidad de clubes. Para esto se propone realizar las siguientes funciones y procedimientos (aunque pueden requerirse o ser útil algunos adicionales):

Ejercicio a) La función `hayMotoquerosActivosSinClub()` que describe Verdadero si en algún lugar de la convención hay algún motoquero activo que no tenga todavía un club.

Ejercicio b) El procedimiento `IrAMotoqueroActivoSinClub()` que asumiendo que hay un motoquero activo sin club, deja el cabezal en el mismo.

Ejercicio c) El procedimiento `AsignarClub_AMotoquerosConectados(númeroDeClub)` que marca todos los motoqueros conectados al motoquero actual, y el actual, como pertenecientes al club con número "númeroDeClub". Dichos motoqueros no deben tener un club previamente asignado. Este problema es a su vez complejo, y se propone la siguiente estrategia para solucionarlo. Paso 1, marcar al motoquero actual como perteneciente al club, y a sus vecinos activos. Paso 2, buscar en la convención algún motoquero activo sin club que tenga como vecino un motoquero del club "númeroDeClub" e ir hasta el mismo, repetir sucesivamente 1 y 2 hasta que no hayan más motoqueros activos en la convención que sean vecinos de otros con el club "númeroDeClub".

Ejercicio d) El procedimiento `AsignarClub_AVecinos(númeroDeClub)` que marca todos los motoqueros vecinos al motoquero actual como pertenecientes al club número "númeroDeClub".

Ejercicio e) La función **hayMotoqueroActivoConVecinoEn_(númeroDeClub)** que retorna Verdadero si en la convención hay algún motoquero activo sin club todavía que tenga de vecino otro motoquero activo que tenga por club “númeroDeClub”.

Ejercicio f) El procedimiento **IrHastaElMotoqueroActivoConVecinoEn_(númeroDeClub)** que posiciona el cabezal sobre un motoquero activo que no tenga club todavía y que tenga por vecino un motoquero en el club “númeroDeClub”

Ejercicio 6)

Escribir el procedimiento **ClubConMásMotoqueros()** que determina cuál es el club que tiene más motoqueros activos en la convención. Se propone como estrategia marcar primero los clubes, y luego contabilizar la cantidad de miembros de cada club, para luego saber cuál tiene más. De esto surgen los siguientes procedimientos y funciones (Nuevamente, realizar adicionales puede resultar útil):

Ejercicio a) El procedimiento **AsignarClubATodosLosMotoquerosActivos()** que asigna un club a cada motoquero activo de la convención.

Ejercicio b) El procedimiento **IrAAlgúnMotoqueroDelClub(númeroDeClub)** que posiciona el cabezal en algún motoquero activo miembro del club “númeroDeClub”.

Ejercicio c) La función **cantidadDeMiembrosDelClubActual()** que determina la cantidad de miembros que posee el club del cual el motoquero actual es miembro. El cabezal debe encontrarse sobre un motoquero que tenga un club asignado.