

# BUSCANDO LAS SEMILLAS

## Objetivos

El ejercicio tiene como objetivo que el estudiante desarrolle un programa que le permita aplicar los siguientes temas de forma conjunta:

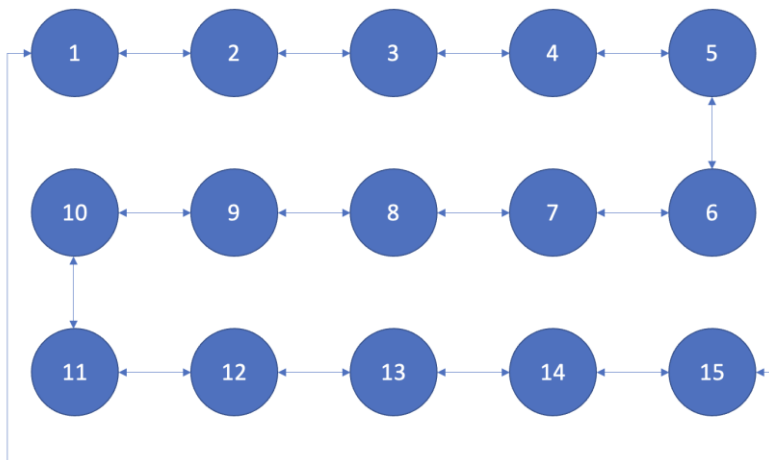
1. Algoritmos de ordenamiento
2. Algoritmos de búsqueda
3. Algoritmos recursivos
4. Listas enlazadas simples, dobles y circulares

## Enunciado

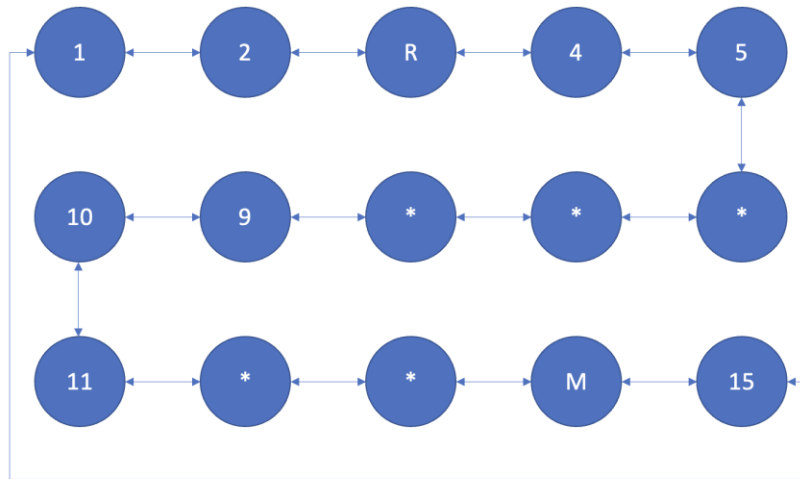


Rick necesita seguir con su investigación sobre la bomba de neutrinos. Para esto necesita recolectar las súper semillas que crecen de los súper árboles que tienen la propiedad de otorgar súper inteligencia temporal. Para recolectar la mayor cantidad posible, Rick se lleva a Morty consigo a la dimensión C-35 gracias a su pistola de portales que permite teletransportarse de un lugar a otro.

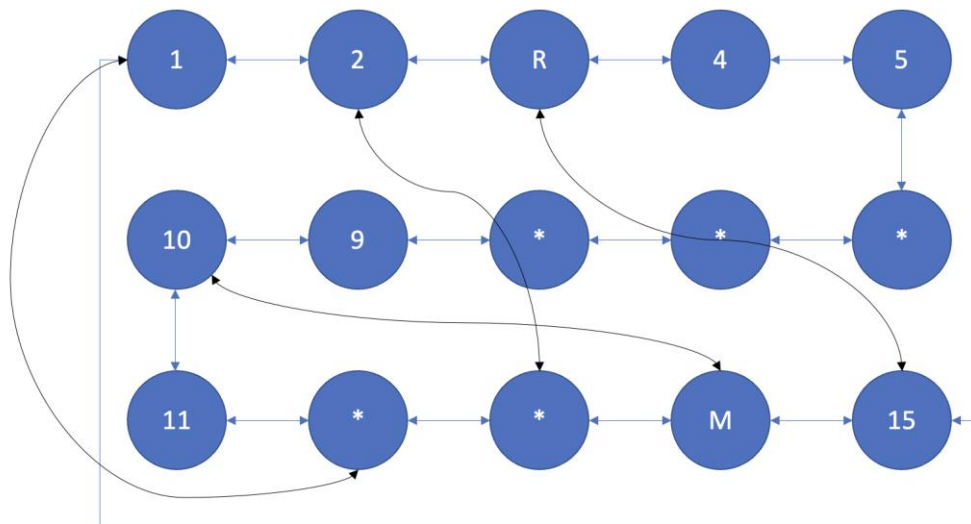
La dimensión a donde llegan puede verse como un tablero  $N \times M$ , donde **N** es el número de columnas y **M** el número de filas con casillas doblemente enlazadas, como se muestra en la siguiente imagen:



En esta dimensión, Rick (R) y Morty (M) son posicionados al azar en el tablero, así como las super semillas (\*).



Para acelerar la recolección, Rick usa su pistola de portales de modo que enlaza **P** pares de casillas del tablero. Quedando entonces un escenario similar al de la siguiente imagen.



En la imagen se puede ver que Rick (R) se encuentra en la casilla 3 y Morty (M) se encuentra en la casilla 14, hay súper semillas en las casillas 6, 7, 8, 12 y 13. Y los portales enlazan las casillas: (1, 12), (2, 13), (10, 14) y (3, 15). Una vez se ha creado el escenario, Rick y Morty pueden empezar a recolectar las súper semillas.

Desarrolle un programa tipo juego por turnos en consola que permita a dos usuarios competir por quién recolecta más semillas.

### Flujo de programa

1. El programa inicialmente pedirá por consola N columnas, M filas y Q semillas, para la creación del tablero. El programa creará la estructura enlazada y pondrá al azar las semillas en el tablero.
2. Luego el programa pide el número de enlaces P. Internamente debe crear esos enlaces en la estructura enlazada del paso 1. Tenga en cuenta que P debe ser un número menor a  $0,5 \cdot (N \cdot M)$ . Evite que una casilla se enlace con 2 o más casillas. Una casilla solo debería estar enlazada con 0 o 1 casilla. Los pares de casillas enlazadas deben ser aleatorios.
3. Rick y Morty serán ubicados al azar en alguna de las casillas del tablero.

**Algoritmos y Programación II**  
**Facultad de ingeniería**

4. Finalmente, el programa pide los usernames de los dos jugadores. Primero el que jugará como Rick, luego el que jugará como Morty.

Ya con todas las condiciones iniciales puede comenzar el juego. Rick comienza jugando.

5. Ya en el juego, primero tirará el dado Rick. El programa la mostrará un menú:

Es el turno de Rick!. ¿Qué deseas hacer?

1. Tirar dado
2. Ver tablero
3. Ver enlaces
4. Marcador

6. Si el jugador pide ver el tablero usando la opción 2, se le muestra una representación en ASCII del tablero. Por ejemplo:

```
[1] [2] [R] [4] [5]
[10] [9] [*] [*] [*]
[11] [*] [*] [M] [15]
```

7. Si el jugador escoge ver los enlaces, aparecerá una representación en ascii de los enlaces

```
[A] [B] [C] [ ] [ ]
[D] [ ] [ ] [ ] [ ]
[ ] [A] [B] [D] [C]
```

Use letras para representar el par de casillas enlazadas.

8. Si el jugador pide ver el Marcador, se muestra las semillas recolectadas por Rick y por Morty:

Rick: 3 semillas

Morty: 4 semillas

9. Si el jugador tira el dado, el programa pregunta por consola si avanza o retrocede. Considere los siguientes ejemplos de las situaciones que pueden suceder:

> Si Rick saca 5 en la primera jugada, el programa le preguntará si avanza o retrocede. Si avanza, queda en la casilla 8. En esta casilla hay una super semilla, así que la recolecta y suma, quedando el tablero así:

```
[1] [2] [3] [4] [5]
[10] [9] [R] [*] [*]
[11] [*] [*] [M] [15]
```

> Si Morty saca 1, el programa le preguntará si avanza o retrocede. Si retrocede, Morty va a la casilla 13 donde hay una semilla, la recolecta. En esa casilla también hay un portal que dirige a la casilla 2. Así que Morty queda en la casilla 2. El tablero queda así:

```
[1] [M] [3] [4] [5]
[10] [9] [R] [*] [*]
[11] [*] [13] [14] [15]
```

> Si en algún momento Rick o Morty caen en una casilla donde hay una semilla y un portal, la semilla es recolectada por el personaje y se teletransporta a la casilla a donde dirija el portal. Si está casilla a donde llegaron tiene una semilla, la semilla se recolecta también, habiendo obtenido 2 semillas en un solo turno.

El juego termina cuando todas las semillas hayan sido recolectadas, luego se mostrará un tablero de resultados

10. Al final del juego sólo se tendrá en cuenta al jugador ganador (quien tenga más semillas). Se calcula su puntaje usando la siguiente formula:

puntaje del ganador = semillas recolectadas \* 120 - tiempo en segundos

El programa mostrará el resultado por ejemplo:

Morty ha ganado recolectando 6 semillas

11. Luego, el programa muestra el TOP 5. Sólo el ganador puede entrar al tablero de resultados. Cada uno de los puntajes deben ir a un archivo que almacena el nombre del jugador ganador y el puntaje obtenido.

Los usuarios pueden acumular el puntaje. Si por ejemplo el jugador **ivan123** jugó una primera vez y ganó haciendo 1000 puntos, si vuelve a jugar y ganar haciendo 2000 puntos, el puntaje de ese username se acumulará dejando a ivan123 con 3000 puntos.

#### Condiciones para el tablero

El tablero DEBE ser una estructura enlazada.

SOLO use recursividad para hacer los recorridos a través del tablero. No se permiten bucles para recorrer el tablero.

#### Condiciones para el tablero de resultados

El tablero de resultados DEBE usar arreglos, colecciones o listas para poder usar los métodos de ordenamiento y búsqueda.

Los puntajes se DEBEN almacenar en un archivo de modo que el histórico de puntajes no se pierda.

#### Calificación

#### Entregables:

**[10%] Especificación de Requerimientos Funcionales.** Liste los requerimientos funcionales del programa. Para esto puede dividir los requerimientos por categorías y numere adecuadamente los requerimientos.

Cada uno puede ser un párrafo donde se describa el requerimiento, por ejemplo:

AUTH1: El empleado debe poder identificarse en el sistema por medio del inicio de sesión. El empleado tuvo que ser registrado desde antes para que el programa le permitiera el ingreso.

Aquí la nomenclatura AUTH hace referencia al módulo de autenticación y el 1 a la numeración.

**[10%] Diagrama de clases**

**[80%] Solución implementada**