



# **Algoritmos y Programación II**

## **Trabajo Práctico N 4: “Nanotecnología”**

## **Objetivo**

Desarrollar un programa que, basándose en el Trabajo Practico N1, N2 y N3, cree los objetos de entrada, los visualice dentro del entorno de juego, permita el uso del simulador Nanobot y agregue la funcionalidad de transportarse hacia una célula determinada con el menor costo posible.

## **Descripción**

Debemos acelerar el tratamiento para la cura del virus Z! Para ello, cada filamento intermedio en forma de cuerda dentro de la red tendrá asociado un peso que representa el costo de recorrer dicho filamento.

Nuestro objetivo es permitir que dentro del simulador descrito en los TPs anteriores, el Nanobot pueda moverse de forma automática hacia una célula determinada utilizando el camino mínimo (suma mínima de pesos de los filamentos) para poder curarla. Respetando la siguiente configuración de teclas:

Tecla C + nro. célula: desde la ubicación, el Nanobot calcula el camino mínimo a la célula ingresada y automáticamente se mueve hacia ella para curarla.

## **Implementación base**

Se les entregó el paquete que contiene la clase Entorno. Esta clase es capaz de encargarse de la interfaz gráfica y de la interacción con el usuario. Así, el grupo solo tendrá que encargarse de la implementación de la “inteligencia” del juego.

## Datos de entrada

Se deberá leer el archivo de texto similar a la forma descripta en el TP N 1 pero con algunas modificaciones / agregados: los filamentos, con sus respectivos pesos y células que conectan, estarán luego de todas las células, anticuerpos y dosis. Cada célula tendrá, además de los datos que ya tenía: tipo y posición, un identificador numérico. Los filamentos, tendrán dos identificadores: las células que conectan, además de su peso.

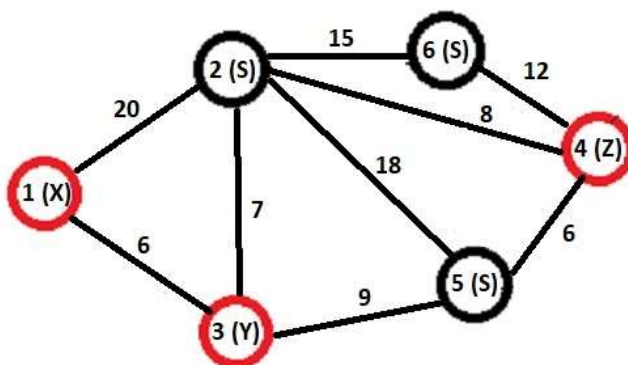
Ejemplo:

```
Célula 1 X 10.5 10.3
celula 2 S 100.2 70.4
celula 3 Y 200 105.3
anticuerpo 500.32 452.4
celula 4 Z 340.2 354.5
celula 5 S 504.13 235.6
anticuerpo 765.4 467.8
dosis B 15
dosis A 15
celula 6 S 310 235
filamento 1 2 20
filamento 1 3 6
filamento 2 3 7
filamento 3 5 9
filamento 2 5 18
filamento 2 6 15
filamento 2 4 8
filamento 4 6 12
filamento 4 5 6
```

La primera línea nos indica que hay una célula de tipo X en la coordenada (10.5;10.3) con identificador 1, luego tenemos una célula de tipo S con identificador 2, etc. Los anticuerpos y las dosis no tienen id.

Luego leemos un filamento que conecta a la célula 1 con la 2 y tiene peso 20, otro que conecta la 1 con la 3, de peso 6, etc.

Entonces, el grafo asociado al archivo del ejemplo quedaría como sigue:



## **Salida**

La aplicación debe, luego de procesar el archivo de entrada, crear todos los objetos involucrados, ubicándolos al momento de su creación en las posiciones indicadas dentro del escenario de juego y permitir la simulación tal como se describió.

## **Aclaraciones**

- Las mismas consideradas en el TP N 1, 2 y 3.
- El archivo está bien formado (no tiene errores).

## **Normas de entrega**

Se deberá subir un archivo comprimido al campus, en un link que se habilitará para esta entrega. Este archivo deberá tener un nombre formado de la siguiente manera:

*NombreGrupo-TP4*

Deberá contener los archivos fuente, un archivo entrada.txt de ejemplo y la documentación producida.

**La fecha de entrega vence el día lunes 2/12/2019 a las 23.55hs.**

Se evaluará: funcionalidad, eficiencia, algoritmo utilizado, buenas prácticas de programación (como nombres, sangrías, etc.), modularización y comentarios.

## **Presentación y defensa**

Entre el martes 3 y el jueves 5 cada grupo hará una presentación del tp3 y el tp4 a alguno de los docentes del curso. La presentación consiste en una muestra completa del funcionamiento, la explicación de la documentación y del código. Además deberán

correrlo con Valgrind para verificar si hay o no pérdida de memoria. Cada integrante deberá explicar cuál fue su aporte al trabajo práctico, por lo que en el momento de la defensa deben estar todos los integrantes del grupo.

**Puntaje:** 25 puntos.