Universidad del Istmo de Guatemala
Facultad de Ingenieria
Ing. en Sistemas
Informatica 1
Prof. Ernesto Rodriguez - erodriguez@unis.edu.gt

Hoja de trabajo #1

Samantha Rodas Chaluleu
Adalí Garrán Jiménez
30 de Julio, 2019

Ejercicio 1

Realizado en GitHub

Ejercicio 2: Abstracción

1. Conjunto de Nodos

(1,1)	(2,1)	(3,1)	(4,1)	(5,1)	(6,1)
(1,2)	(2,2)	(3,2)	(4,2)	(5,2)	(6,2)
(1,3)	(2,3)	(3,3)	(4,3)	(5,3)	(6,3)
(1,4)	(2,4)	(3,4)	(4,4)	(5,4)	(6,4)
(1,4)	(2,5)	(3,5)	(4,5)	(5,5)	(6,5)
(1,6)	(2,6)	(3,6)	(4,6)	(5,6)	(6,6)

2. Conjunto de Vértices

<(1,1),(1,2)>, <(1,1),(2,1)>	<(1,2),(1,1)>, <(1,2),(2,2)>	<(1,3),(1,1)>, <(1,3),(2,3)>
<(1,1),(1,3)>, <(1,1),(3,1)>	<(1,2),(1,3)>, <(1,2),(3,2)>	<(1,3),(1,2)>, <(1,3),(3,3)>
<(1,1),(1,4)>, <(1,1),(4,1)>	<(1,2),(1,4)>, <(1,2),(4,2)>	<(1,3),(1,5)>, <(1,3),(4,3)>
<(1,1),(1,5)>, <(1,1),(5,1)>	<(1,2),(1,6)>, <(1,2),(5,2)>	<(1,3),(1,6)>, <(1,3),(5,3)>
<(1,4),(1,1)>, <(1,4),(2,4)>	<(1,5),(1,1)>, <(1,5),(2,5)>	<(1,6),(1,2)>, <(1,6),(2,6)>
<(1,4),(1,2)>, <(1,4),(3,4)>	<(1,5),(1,3)>, <(1,5),(3,5)>	<(1,6),(1,3)>, <(1,6),(3,6)>
<(1,4),(1,5)>, <(1,4),(4,4)>	<(1,5),(1,4)>, <(1,5),(4,5)>	<(1,6),(1,4)>, <(1,6),(4,6)>
<(1,4),(1,6)>, <(1,4),(5,4)>	<(1,5),(1,6)>, <(1,5),(5,5)>	<(1,6),(1,5)>, <(1,6),(5,6)>
<(2,1),(2,2)>, <(2,1),(1,1)>	<(2,2),(2,1)>, <(2,2),(1,2)>	<(2,3),(2,1)>, <(2,3),(1,3)>
<(2,1),(2,3)>, <(2,1),(3,1)>	<(2,2),(2,3)>, <(2,2),(3,2)>	<(2,3),(2,2)>, <(2,3),(3,3)>
<(2,1),(2,4)>, <(2,1),(4,1)>	<(2,2),(2,4)>, <(2,2),(4,2)>	<(2,3),(2,5)>, <(2,3),(4,3)>
<(2,1),(2,5)>, <(2,1),(6,1)>	<(2,2),(2,6)>, <(2,2),(6,2)>	<(2,3),(2,6)>, <(2,3),(6,3)>
<(2,4),(2,1)>, <(2,4),(1,4)>	<(2,5),(2,1)>, <(2,5),(1,5)>	<(2,6),(2,2)>, <(2,6),(1,6)>
<(2,4),(2,2)>, <(2,4),(3,4)>	<(2,5),(2,3)>, <(2,5),(3,5)>	<(2,6),(2,3)>, <(2,6),(3,6)>

<(2,4),(2,5)>, <(2,4),(4,4)>	<(2,5),(2,4)>, <(2,5),(4,5)>	<(2,6),(2,4)>, <(2,6),(4,6)>
<(2,4),(2,6)>, <(2,4),(6,4)>	<(2,5),(2,6)>, <(2,5),(6,5)>	<(2,6),(2,5)>, <(2,6),(6,6)>
<(3,1),(3,2)>, <(3,1),(1,1)>	<(3,2),(3,1)>, <(3,2),(1,2)>	<(3,3),(3,1)>, <(3,3),(1,3)>
<(3,1),(3,3)>, <(3,1),(2,1)>	<(3,2),(3,3)>, <(3,2),(2,2)>	<(3,3),(3,2)>, <(3,3),(2,3)>
<(3,1),(3,4)>, <(3,1),(5,1)>	<(3,2),(3,4)>, <(3,2),(5,2)>	<(3,3),(3,5)>, <(3,3),(5,3)>
<(3,1),(3,5)>, <(3,1),(6,1)>	<(3,2),(3,6)>, <(3,2),(6,2)>	<(3,3),(3,6)>, <(3,3),(6,3)>
<(3,4),(3,1)>, <(3,4),(1,4)>	<(3,5),(3,1)>, <(3,5),(1,5)>	<(3,6),(3,2)>, <(3,6),(1,6)>
<(3,4),(3,2)>, <(3,4),(2,4)>	<(3,5),(3,3)>, <(3,5),(2,5)>	<(3,6),(3,3)>, <(3,6),(2,6)>
<(3,4),(3,5)>, <(3,4),(5,4)>	<(3,5),(3,4)>, <(3,5),(5,5)>	<(3,6),(3,4)>, <(3,6),(5,6)>
<(3,4),(3,6)>, <(3,4),(6,4)>	<(3,5),(3,6)>, <(3,5),(6,5)>	<(3,6),(3,5)>, <(3,6),(6,6)>
<(4,1),(4,2)>, <(4,1),(1,1)>	<(4,2),(4,1)>, <(4,2),(1,2)>	<(4,3),(4,1)>, <(4,3),(1,3)>
<(4,1),(4,3)>, <(4,1),(2,1)>	<(4,2),(4,3)>, <(4,2),(2,2)>	<(4,3),(4,2)>, <(4,3),(2,3)>
<(4,1),(4,4)>, <(4,1),(5,1)>	<(4,2),(4,4)>, <(4,2),(5,2)>	<(4,3),(4,5)>, <(4,3),(5,3)>
<(4,1),(4,5)>, <(4,1),(6,1)>	<(4,2),(4,6)>, <(4,2),(6,2)>	<(4,3),(4,6)>, <(4,3),(6,3)>
<(4,4),(4,1)>, <(4,4),(1,4)>	<(4,5),(4,1)>, <(4,5),(1,5)>	<(4,6),(4,2)>, <(4,6),(1,6)>
<(4,4),(4,2)>, <(4,4),(2,4)>	<(4,5),(4,3)>, <(4,5),(2,5)>	<(4,6),(4,3)>, <(4,6),(2,6)>
<(4,4),(4,5)>, <(4,4),(5,4)>	<(4,5),(4,4)>, <(4,5),(5,5)>	<(4,6),(4,4)>, <(4,6),(5,6)>
<(4,4),(4,6)>, <(4,4),(6,4)>	<(4,5),(4,6)>, <(4,5),(6,5)>	<(4,6),(4,5)>, <(4,6),(6,6)>
<(5,1),(5,2)>, <(5,1),(1,1)>	<(5,2),(5,1)>, <(5,2),(1,2)>	<(5,3),(5,1)>, <(5,3),(1,3)>
<(5,1),(5,3)>, <(5,1),(3,1)>	<(5,2),(5,3)>, <(5,2),(3,2)>	<(5,3),(5,2)>, <(5,3),(3,3)>
<(5,1),(5,4)>, <(5,1),(4,1)>	<(5,2),(5,4)>, <(5,2),(4,2)>	<(5,3),(5,5)>, <(5,3),(4,3)>
<(5,1),(5,5)>, <(5,1),(6,1)>	<(5,2),(5,6)>, <(5,2),(6,2)>	<(5,3),(5,6)>, <(5,3),(6,3)>
<(5,4),(5,1)>, <(5,4),(1,4)>	<(5,5),(5,1)>, <(5,5),(1,5)>	<(5,6),(5,2)>, <(5,6),(1,6)>
<(5,4),(5,2)>, <(5,4),(3,4)>	<(5,5),(5,3)>, <(5,5),(3,5)>	<(5,6),(5,3)>, <(5,6),(3,6)>
<(5,4),(5,5)>, <(5,4),(4,4)>	<(5,5),(5,4)>, <(5,5),(4,5)>	<(5,6),(5,4)>, <(5,6),(4,6)>
<(5,4),(5,6)>, <(5,4),(6,4)>	<(5,5),(5,6)>, <(5,5),(6,5)>	<(5,6),(5,5)>, <(5,6),(6,6)>
<(6,1),(6,2)>, <(6,1),(2,1)>	<(6,2),(6,1)>, <(6,2),(2,2)>	<(6,3),(6,1)>, <(6,3),(2,3)>
<(6,1),(6,3)>, <(6,1),(3,1)>	<(6,2),(6,3)>, <(6,2),(3,2)>	<(6,3),(6,2)>, <(6,3),(3,3)>
<(6,1),(6,4)>, <(6,1),(4,1)>	<(6,2),(6,4)>, <(6,2),(4,2)>	<(6,3),(6,5)>, <(6,3),(4,3)>
<(6,1),(6,5)>, <(6,1),(5,1)>	<(6,2),(6,6)>, <(6,2),(5,2)>	<(6,3),(6,6)>, <(6,3),(5,3)>
<(6,4),(6,1)>, <(6,4),(2,4)>	<(6,5),(6,1)>, <(6,5),(2,5)>	<(6,6),(6,2)>, <(6,6),(2,6)>
<(6,4),(6,2)>, <(6,4),(3,4)>	<(6,5),(6,3)>, <(6,5),(3,5)>	<(6,6),(6,3)>, <(6,6),(3,6)>
<(6,4),(6,5)>, <(6,4),(4,4)>	<(6,5),(6,4)>, <(6,5),(4,5)>	<(6,6),(6,4)>, <(6,6),(4,6)>
<(6,4),(6,6)>, <(6,4),(5,4)>	<(6,5),(6,6)>, <(6,5),(5,5)>	<(6,6),(6,5)>, <(6,6),(5,6)>

Ejercicio 3

Indique como se puede usar el grafo anterior para el lanzamiento de dados empezando en (1,1):

Para representar un lanzamiento de dados según el grafo anterior, se supone que para llegar a la cara de

los dados que se obtiene del lanzamiento, los dados se mueven uno a la vez y solamente rotando 90 grados en cada movimiento.

- ¿Qué estructura de datos podría representar un lanzamiento de dados?

  Grafo, porque los grafos son estructuras de datos conectadas compuestas por nodos. En este caso las combinaciones posibles entre las caras de los dados son los nodos y los movimientos para llegar a ellas son los vértices.

- ¿Qué algoritmo podríamos utilizar para generar dicha estructura de datos?

  Una adaptación del Algoritmo de Dijkstra, porque este es un algoritmo para la determinación del camino más corto, dado un vértice origen, hacia el resto de los vértices en un grafo que tiene pesos en cada arista. Funciona con nuestro lanzamiento ya que el camino se traza por nodos adyacentes, lo que asegura que cada dado se movería 90 grados cada vez. Además se agregan 4 pasos más que aseguran la aleatoriedad del tiro. Adaptando este algoritmo a nuestro problema inicial, este sería el resultado:

  1. Inicializar todas las distancias en las caras del dado con un valor infinito relativo, ya que son desconocidas al principio, exceptuando la de x, que se debe colocar en 0, debido a que la distancia de x a x sería 0.
  2. Sea x=(1,1) el nodo actual o inicial.
  3. Se recorren todos los nodos adyacentes de (1,1), excepto los nodos marcados. Se les llamará nodos no marcados vi.
  4. Para el nodo actual, se calcula la distancia tentativa desde dicho nodo hasta sus vecinos. La distancia tentativa del nodo 'vi' es la distancia que actualmente tiene el nodo en el Dado más la distancia desde dicho nodo '(1,1)' hasta el nodo vi. Si la distancia tentativa es menor que la distancia almacenada, entonces se actualiza.
  5. Se marca como completo el nodo (1,1).
  6. Se toma un siguiente nodo actual y se repite el paso 3.
  7. Se generan dos números aleatorios entre 1 y 6 (pudiéndose repetir).
  8. Se mueve el primer dado hacia su valor correspondiente, según la distancia (dada en el paso 4) desde el nodo actual hasta el primer número generado en el paso 7.
  9. Se mueve el segundo dado hacia su valor correspondiente, según la distancia (dada en el paso 4) desde el nodo actual hasta el segundo número generado en el paso 7.
  10. Se obtiene de las caras expuestas de los dados el resultado correspondiente a los movimientos realizados.

- ¿Cómo nos aseguramos que ese algoritmo siempre produce un resultado?

  Como se mencionó anteriormente, el Algoritmo de Dijkstra busca el camino más corto entre un vértice inicial y los demás vértices de un grafo, el cual posee nodos no aislados. Cada cara de los dados está conectada directamente a otras cuatro caras, y siempre habrá una cara mostrándose hacia arriba. En este caso el nodo inicial es (1,1). Ya que ninguna de las caras está aislada, siempre existen caminos entre ellas y se puede llegar a todos los distintos vértices del grafo. Además, ya que siempre se generarán dos números aleatorios entre 1 y 6 (iguales a las caras de los dados), siempre se obtendrá un resultado.

Ejercicio 4

Realizado en GitHub