

Hoja de trabajo No.1

Sébastien Escobar

July 25, 2018

Informatica 1

Ejercicio No.2 Abstracción

Considerando un dado de seis caras, el cual se coloca en una superficie y solo puede rotar 90 grados en cualquiera de sus 4 lados.

Se hace una modelacion del dado y sus transiciones de estado como un grafo. Para ello se conforma el conjunto de nodos sobre el grafo:

Nodos

$\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}$

Sean los nodos las 6 posibilidades en las cuales podria caer el dado segun el tiro que se haga. Por otro lado los vertices, que se presentan a continuación son las parejas ordenadas por las cuales el dado puede moverse. Hasta llegar a su destino siendo este el nodo final. Y tomando 1 como el inicial.

Vertices

$\langle 1,2 \rangle$ $\langle 2,6 \rangle$ $\langle 6,5 \rangle$ $\langle 5,1 \rangle$
 $\langle 1,4 \rangle$ $\langle 4,6 \rangle$ $\langle 6,3 \rangle$ $\langle 3,1 \rangle$
 $\langle 1,5 \rangle$ $\langle 5,6 \rangle$ $\langle 6,2 \rangle$ $\langle 2,1 \rangle$
 $\langle 1,3 \rangle$ $\langle 3,6 \rangle$ $\langle 6,4 \rangle$ $\langle 4,1 \rangle$

Simulación utilizando un grafo

Ejercicio No.3

¿Que estructura de datos podria representar un lanzamiento de dados?

Un grafo en el cual existe un camino con dos nodos especiales uno de los cuales comienza en "1" y posee dos ciclos que viaja a través de vértices con los cuales se conectan a cualquier posibilidad del dado siendo este el segundo nodo. El camino a cuál sea el número (aleatorio) al cual se quiere llegar, está determinado por un conjunto de vértices antes de llegar al mismo.

¿Que algoritmo podriamos utilizar para generar dicha estructura?

Podríamos utilizar el algoritmo de Búsqueda de Uniones. Ya que se colocan nodos para los que exista un camino en una misma clase de equivalencia. Antes de agregar un vértice $[x,y]$, revisar que X y Y no se encuentren en la misma clase de equivalencia.

¿Como nos aseguramos que ese algoritmo siempre produce un resultado?

Antes de agregar un vertice $[1, x]$, se revisa que x no se encuentre en la misma clase de equivalencia. Se colocan los nodos para los que exista un camino en una misma clase de equivalencia. Por ultimo se encuentran los vertices que mejor se acoplen para llegar eficientemente al segundo nodo. Y así lograr un resultado rapido y eficiente cada vez con este algoritmo.