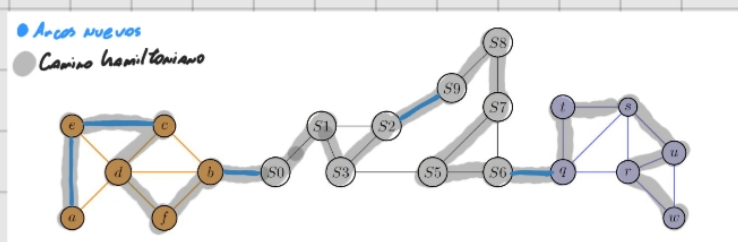


1.- G_1 debe tener un camino hamiltoniano tal que este inicie en a y termine en b , para poder así conectar G_1 con G_2

G_2 debe tener un camino hamiltoniano entre S_0 y S_6

G_3 debe tener un camino hamiltoniano entre q y w

2.- Se deben agregar los siguientes arcos: $\{ (a,e), (e,c), (b,S_0), (S_2,S_4), (S_6,q) \}$, y el camino hamiltoniano generado sería:



que podría ser representado por la siguiente secuencia de vértices:

$\{ a, e, c, d, f, b, S_0, S_1, S_3, S_2, S_4, S_6, q, t, s, u, r, w \}$

Este camino simple cumple con la condición de pasar por todos los vértices sin repetirlos para que el camino sea hamiltoniano

3.-

Algorithm 1 Generar grafos hamiltonianos

```
1: Procedure generarG( $G_1, G_2, G_3 \dots G_n$ )
2:   list grafosConexos  $\leftarrow (G_1, G_2, G_3, \dots G_n)$ 
3:   grafos retorna  $\leftarrow ()$ 
4:   Vertice ultimoVertice CH = ()
5:   for grafos in grafosConexos do
6:     hamiltoniano  $\leftarrow$  FindHamiltonian(grafos)
7:     if hamiltoniano equals NULL then
8:       flag  $\leftarrow$  False
9:       while flag equals False do
10:         vertice1  $\leftarrow$  SearchVertice(grafos)
11:         vertice2  $\leftarrow$  SearchVertice(grafos)
12:         if (vertice1, vertice2) not in grafos do
13:           add((vertice1, vertice2), grafos)
14:         end if
15:         hamiltoniano  $\leftarrow$  FindHamiltonian(grafos)
16:         if hamiltoniano does not equal NULL then
17:           flag  $\leftarrow$  True
18:         end if
19:       end while
20:     end if
21:     add(ultimoVertice CH, hamiltoniano[0], grafos)
22:     add(grafos, retorna)
23:     ult  $\leftarrow$  len(hamiltoniano)
24:     ultimoVertice CH  $\leftarrow$  hamiltoniano[ult-1]
25:   end for
26:   return retorna
```

add() agrega un arco a un grafos, si se le entregan dos

grados, los une.

search Vertex () busca vertices en un grafo de grado menor o igual
A 2