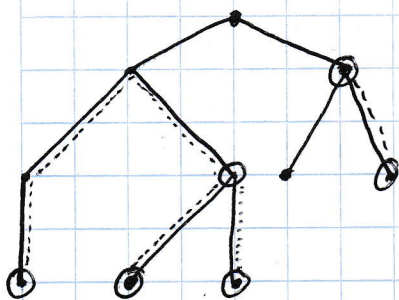


1. Sea T un árbol con raíz r y S un conjunto con un número par de vértices de T . Diseña un algoritmo de tiempo polinomial que encuentre $|S|/2$ caminos simples y disjuntos en aristas que empujen pares de vértices en S . La figura siguiente muestra un ejemplo en el que S consta de los seis vértices enmarcados en círculos y tres caminos (denotados en tres tipos distintos de líneas punteadas) que empujan esos vértices.



```

1 Inicializa ( $n$ )
2   PadreActual =  $\emptyset$  // Variable global que contiene el padre actual
3   VisitadosEnI // Arreglo global, se crea un arreglo de tamaño  $n$  de
4   // valores booleanos
5   CaminosEnI // Arreglo global, se crea un arreglo de tamaño  $n$  donde
6   // cada uno de sus elementos es una cola, es decir es un
7   // arreglo de colas.
8   for  $i=1$  to  $n$ 
9     VisitadoEnI = false
10  endfor
11
12 Conjunto (Adj)
13   Inicializa (Adj.length)
14   if Adj.length > 1 then
15     for  $i=1$  to Adj.length
16       if VisitadosEnI == false && AdjEnI.length > 1
17         emparejado = false // variable global
18         VisitadosEnI = true
19         PadreActual =  $i$ 

```

20 Emparejar_vertice(i , Adj_i)

21 endif

22 endfor

23 endif

24 return Caminos

25

26 Emparejar_vertice(u , Adj_i)

27 if $Adj[u].length > 1$ then

28 for $i=1$ to $Adj[u].length$

29 if $emparejado == false$ \wedge $visitado[Adj[u][i]] == false$ then

30 Emparejar_vertice($Adj[u][i]$, Adj_i)

31 endif

32 if $emparejado == true$ then

33 Eliminar_Relacion(u , $Adj[u][i]$, Adj_i)

34 Eliminar_Relacion($Adj[u][i]$, u , Adj_i)

35 Camino[Padre Actual].encolar(u)

36 endif

37 endfor

38 else

39 emparejado = true

40 visitado[u] = true

41 Camino[Padre Actual].encolar(u)

42 endif

43 endfor

44

45 Eliminar_Relacion(v , u , Adj_i)

46 for $i=1$ to $Adj[v].length$

47 if $Adj[v][i] == u$ then

48 $Adj[v].RemoveElemento(i)$ // elimina el elemento en la posición i

49 endif

50 if $Adj[v].length == \emptyset$ then

51 visitado[v] = true

52 endif


```
if Visitado[i] == true then
    Eliminar_Relacion (Adj[i][i], v, Adj)
    Adj[i].Remove_elemento(i)
endif
endfor
```