Prácticas de Matemática Discreta: Introducción a la teoría de grafos

Sesión 6

Algoritmos de búsqueda

Un algoritmo de búsqueda en un grafo conexo G es un procedimiento sistemático cuyo objetivo es "visitar" todos los vértices de G "viajando" a través de las aristas. Un vértice puede visitarse más de una vez y una arista puede atravesarse más de una vez a lo largo del proceso de búsqueda.

Dicho de otro modo, un algoritmo de búsqueda es un proceso sistemático para construir un subgrafo de G que contiene a todos sus vértices, es decir, un subgrafo generador de G.

Veremos dos algoritmos para construir un tipo particular de subgrafo generador: un árbol generador. Son los siguientes:

- Breadth-first search (o BFS).
- Depth-first search (o DFS).

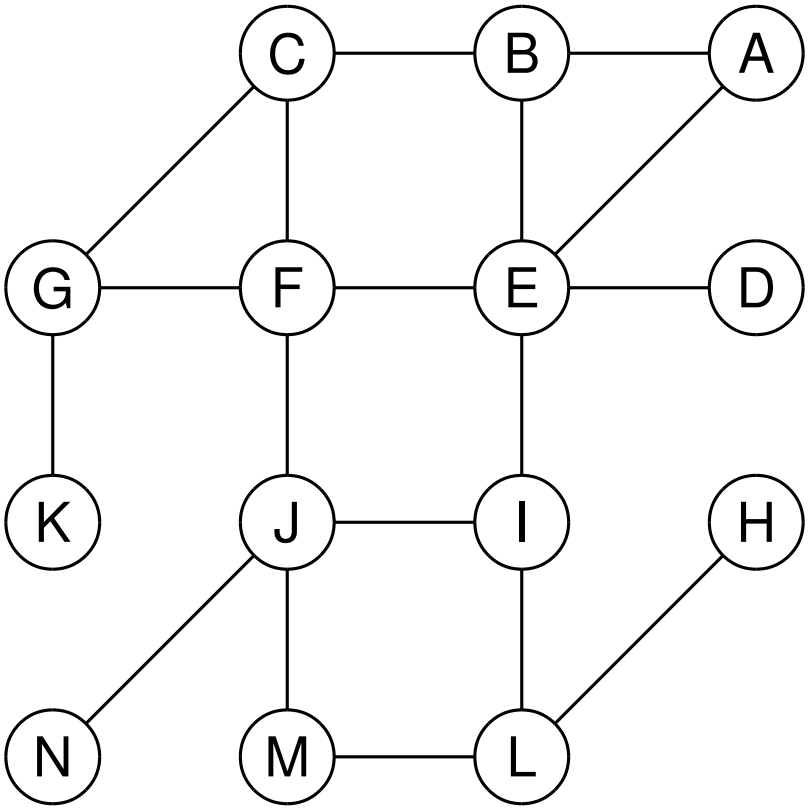
Breadth: amplitud, ancho; depth: profundidad

Algoritmo Breadth-first search (BFS)

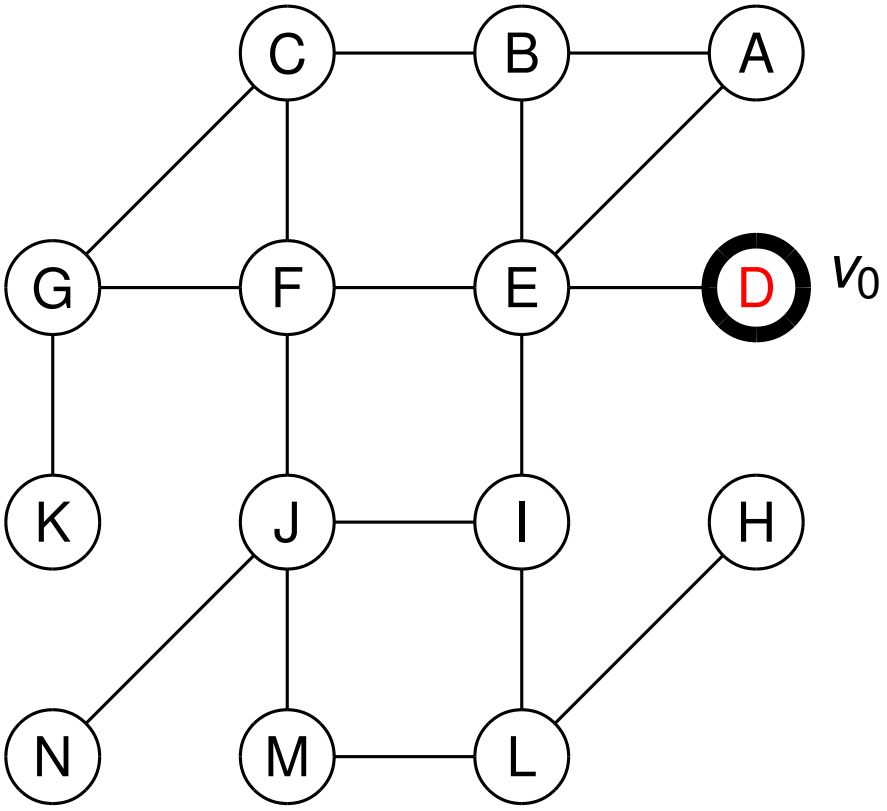
Sea G un grafo conexo.

1. Elige un vértice v_0 del grafo.
2. Sean $m = 0$, $w := v_m = v_0$ y $n = 0$. (Al vértice $w = v_m$ lo denominaremos "centro actual" de la búsqueda). Sea T_0 el árbol sin aristas cuyo único vértice es v_0.
3. Si existe algún vértice nuevo (nuevo significa "que no es un vértice del árbol actual") que sea adyacente a w entonces
   - elige uno de ellos, v_{n+1};
   - añade a T_n el nuevo vértice v_{n+1} así como una arista e_n cuyos extremos sean w y v_{n+1}. Formaremos, así, el siguiente árbol T_{n+1};
   - incrementa n en una unidad;
   - repite el paso 3 hasta que no haya nuevos vértices adyacentes a w.
4. Si todos los vértices han sido visitados entonces T_n es un árbol generador de G y hemos acabado. En otro caso incrementa m en una unidad, toma $w = v_m$ y ve al paso 3.

Ejemplo

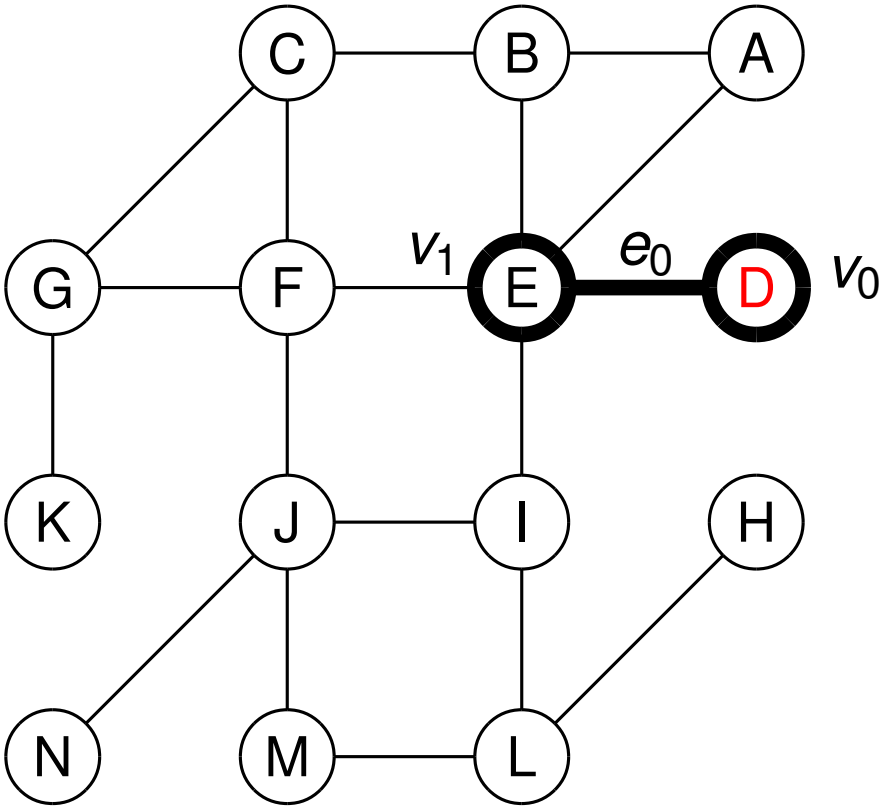


Ejemplo (Pasos 1 y 2)



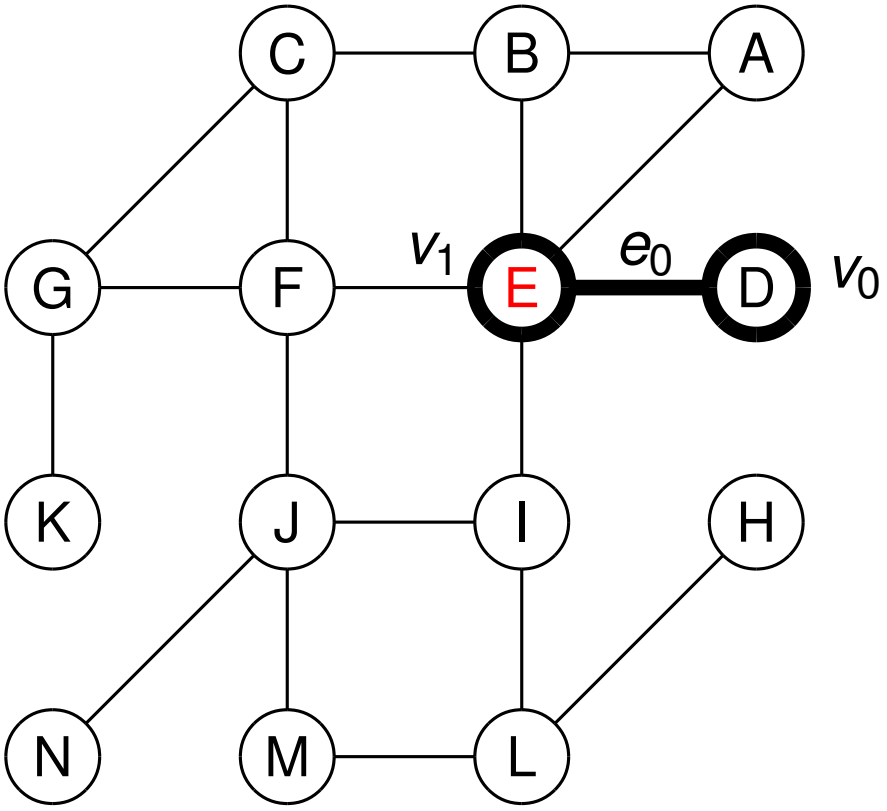
$m = 0$ Centro: $w = v_0 = D$ $n = 0$ T_0 : (árbol dibujado)

Ejemplo (Paso 3)



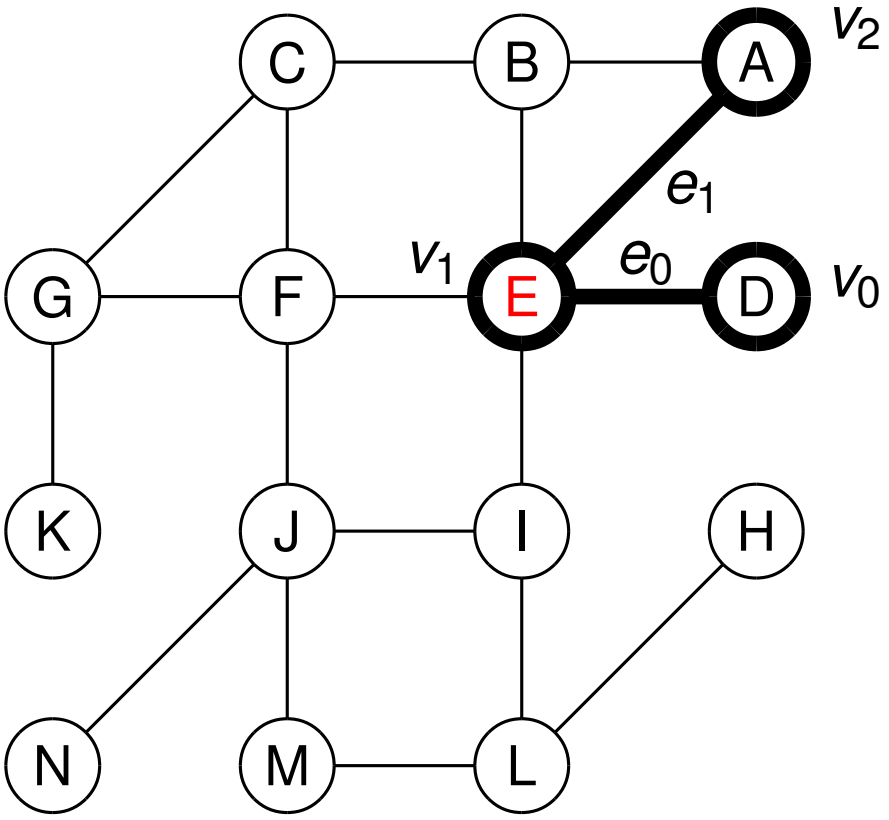
$m = 0$ Centro: $w = v_0 = D$ $n = 1$ T_1 : (árbol dibujado)

Ejemplo (Paso 4)



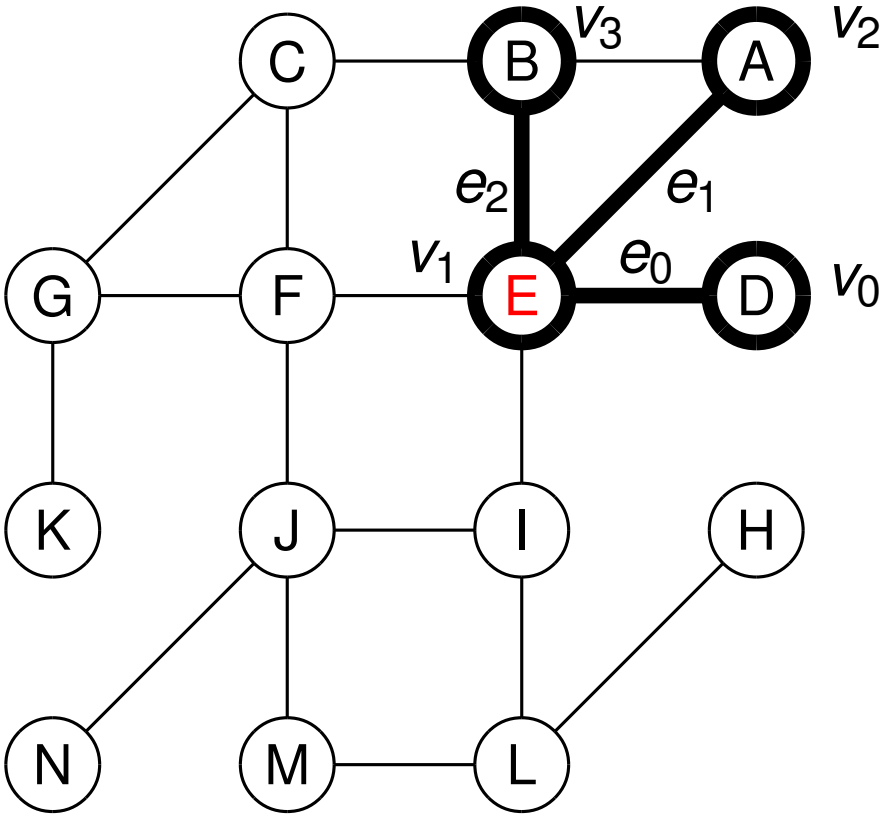
$m = 1$ Centro: $w = v_1 = E$ $n = 1$ T_1 : (árbol dibujado)

Ejemplo (Paso 3)



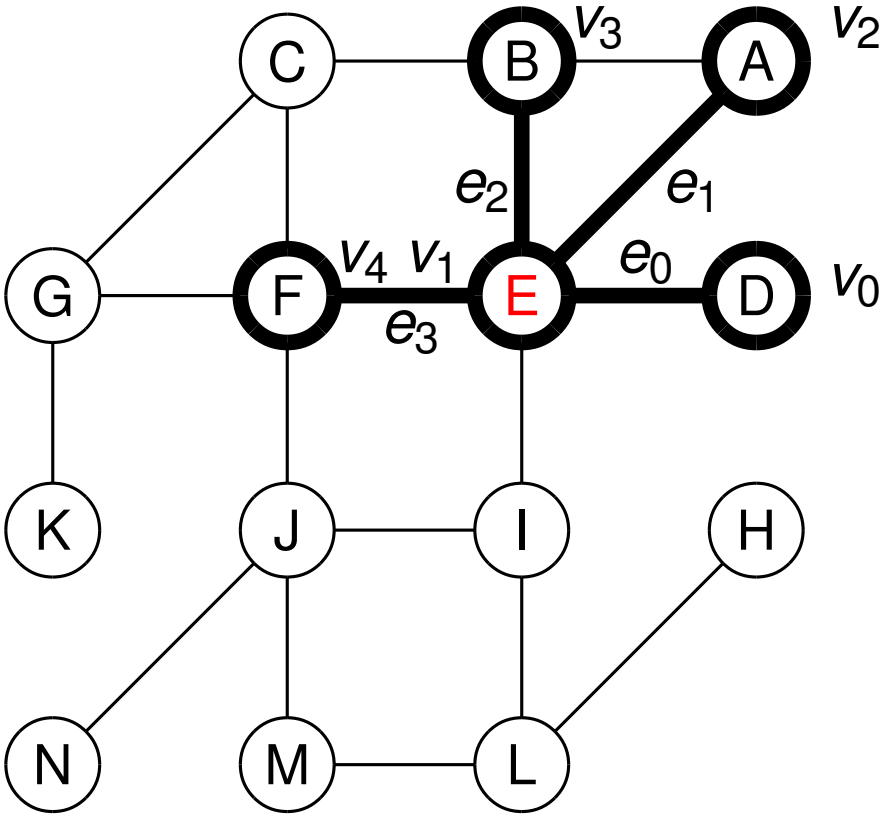
$m = 1$ Centro: $w = v_1 = E$ $n = 2$ T_2 : (árbol dibujado)

Ejemplo (Paso 3)



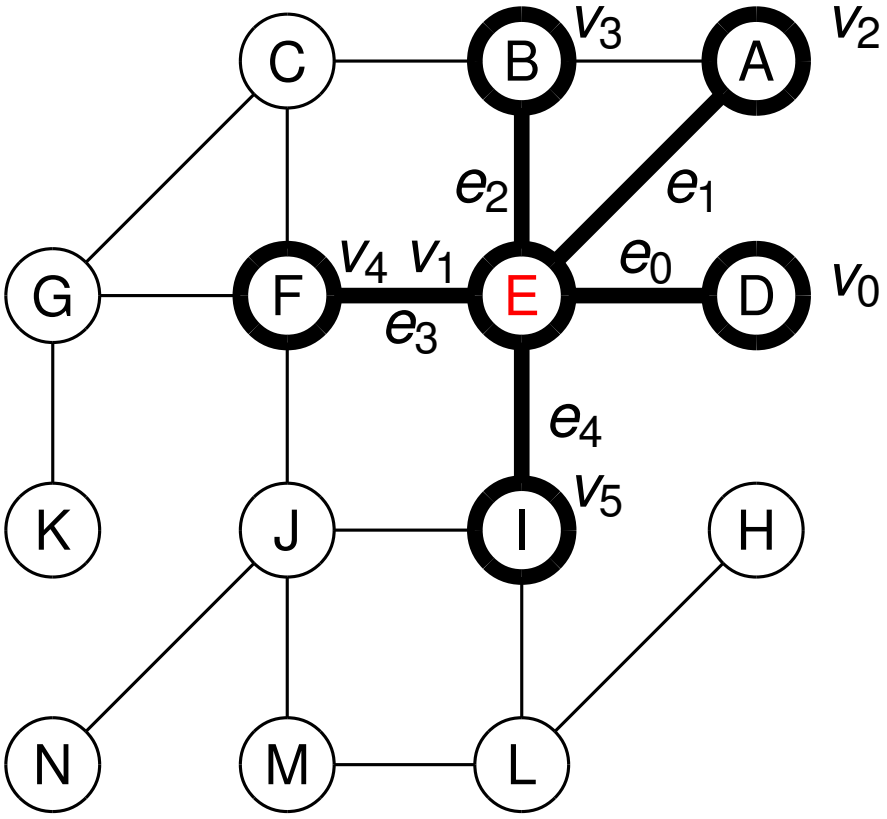
$m = 1$ Centro: $w = v_1 = E$ $n = 3$ T_3 : (árbol dibujado)

Ejemplo (Paso 3)



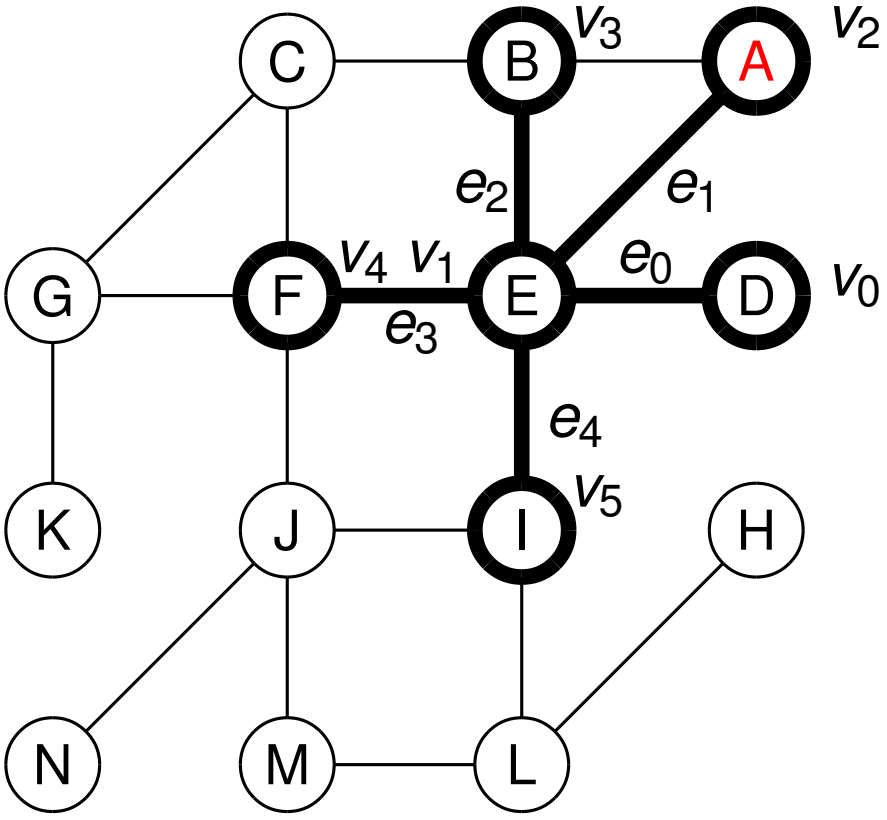
$m = 1$ Centro: $w = v_1 = E$ $n = 4$ T_4 : (árbol dibujado)

Ejemplo (Paso 3)



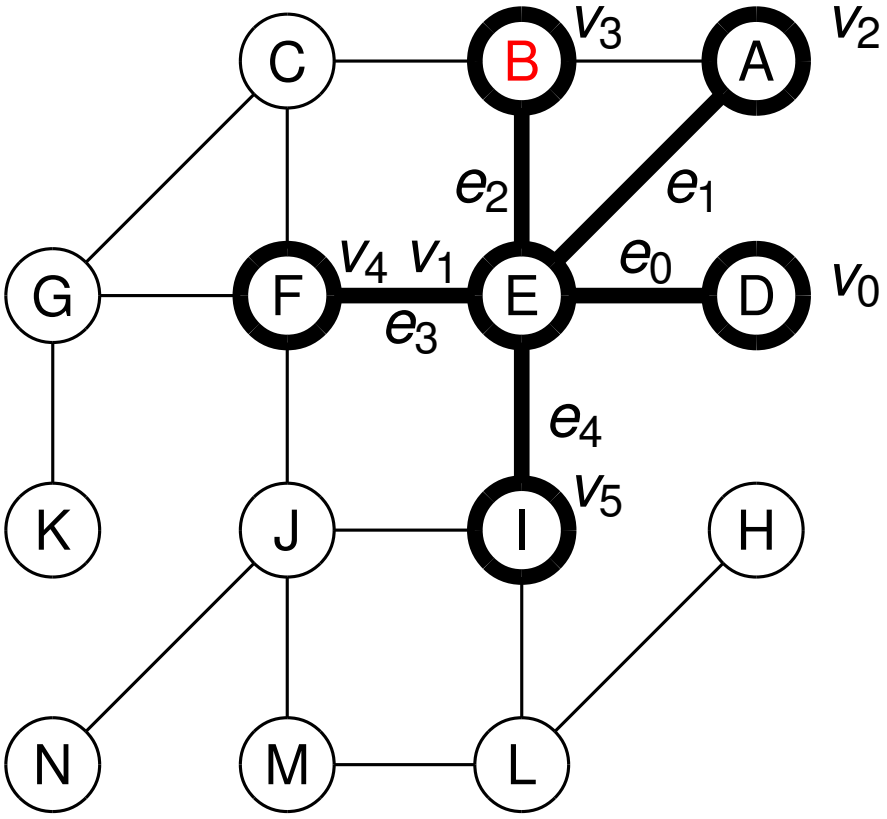
$m = 1$ Centro: $w = v_1 = E$ $n = 5$ T_5 : (árbol dibujado)

Ejemplo (Paso 4)



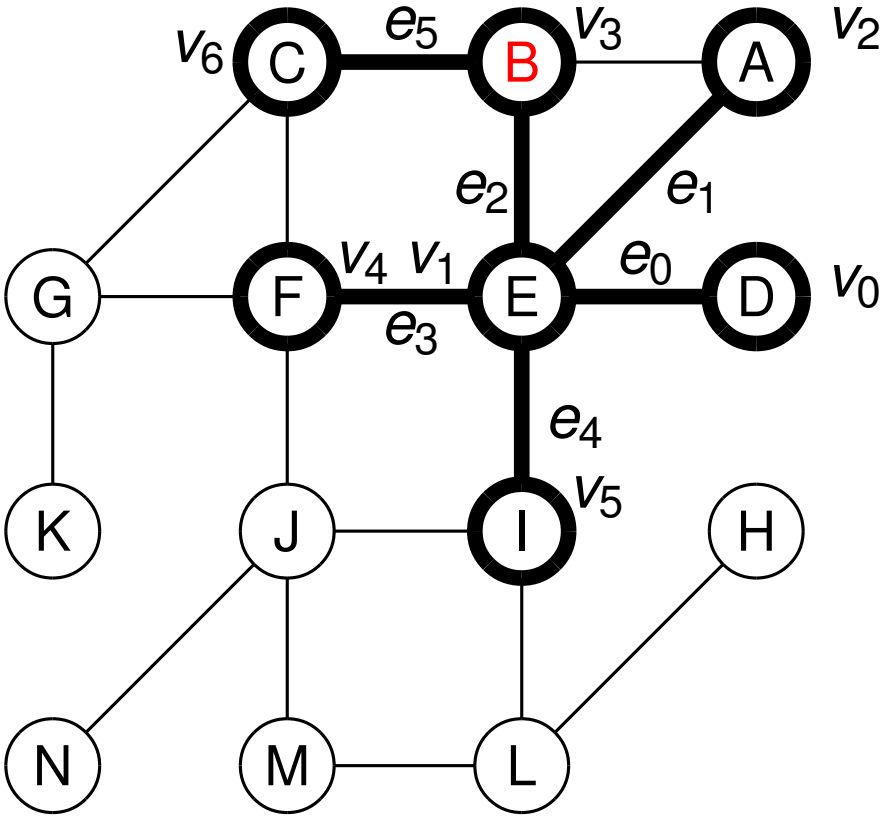
$m = 2$ Centro: $w = v_2 = A$ $n = 5$ T_5 : (árbol dibujado)

Ejemplo (Paso 4)



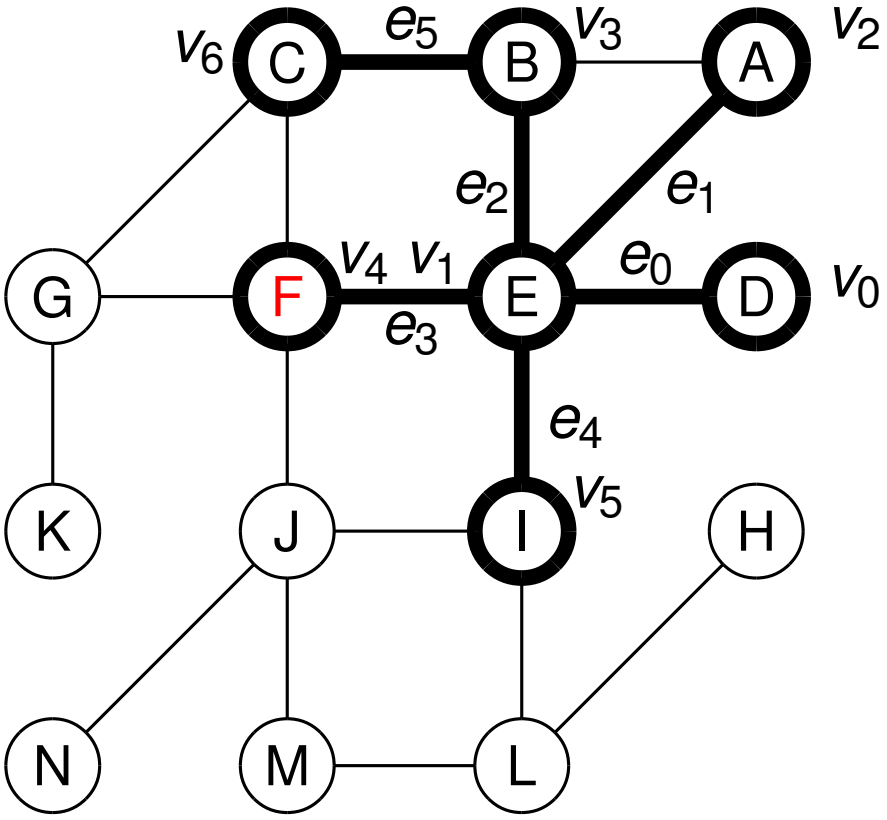
$m = 3$ Centro: $w = v_3 = B$ $n = 5$ T_5 : (árbol dibujado)

Ejemplo (Paso 3)



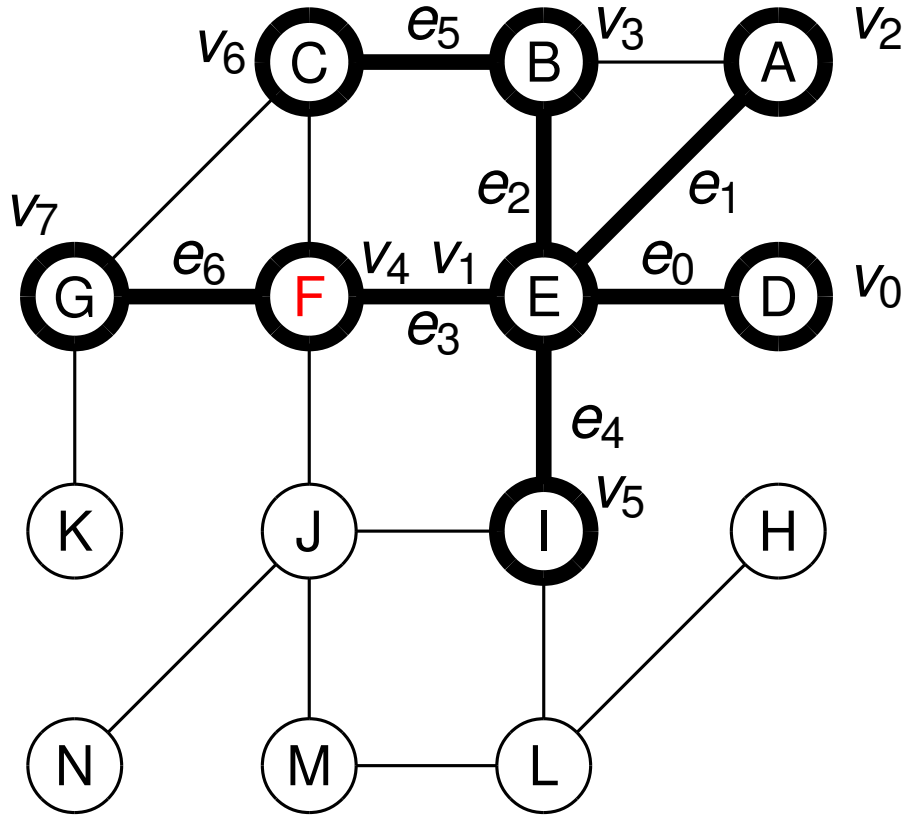
$m = 3$ Centro: $w = v_3 = B$ $n = 6$ T_6 : (árbol dibujado)

Ejemplo (Paso 4)



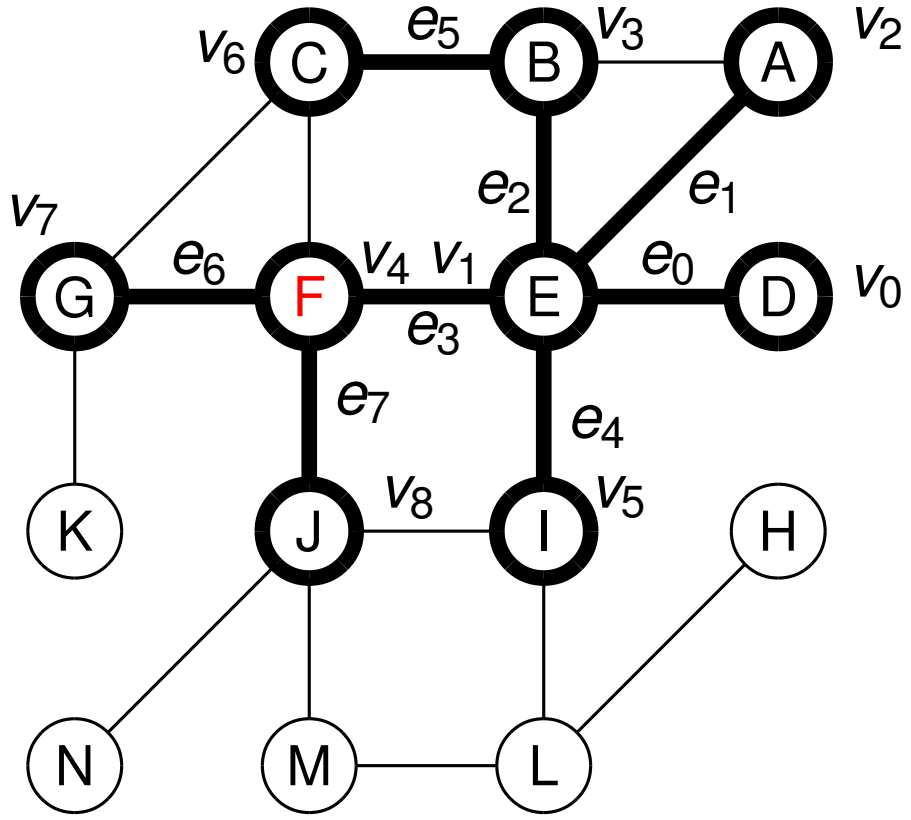
$m = 4$ Centro: $w = v_4 = F$ $n = 6$ T_6 : (árbol dibujado)

Ejemplo (Paso 3)



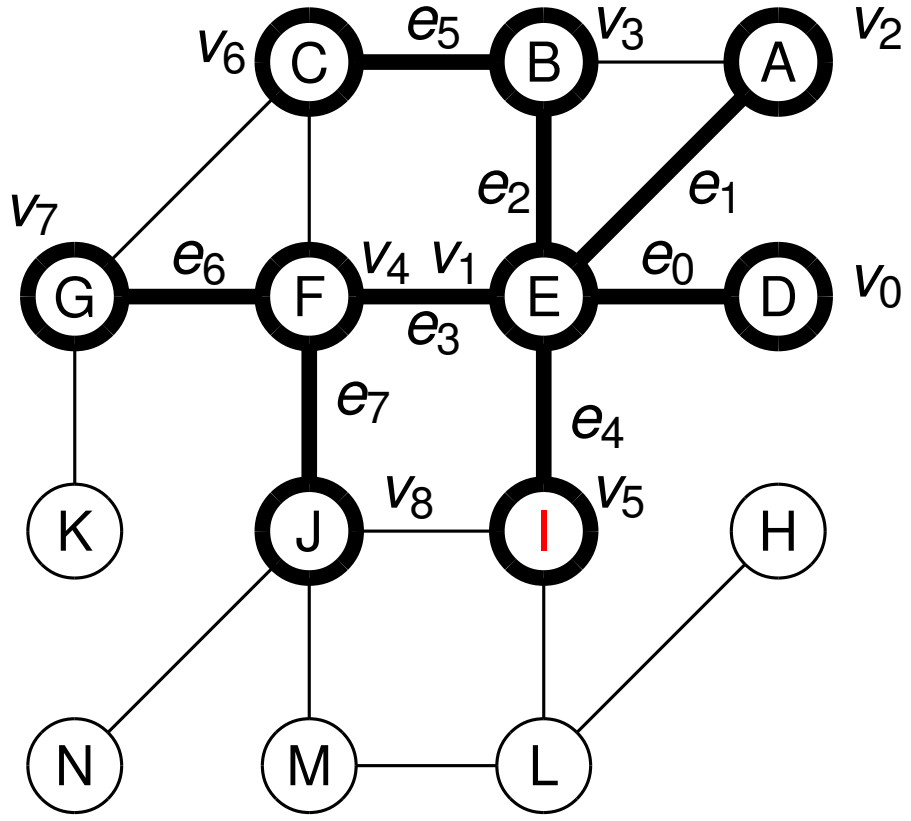
$m = 4$ Centro: $w = v_4 = F$ $n = 7$ T_7 : (árbol dibujado)

Ejemplo (Paso 3)



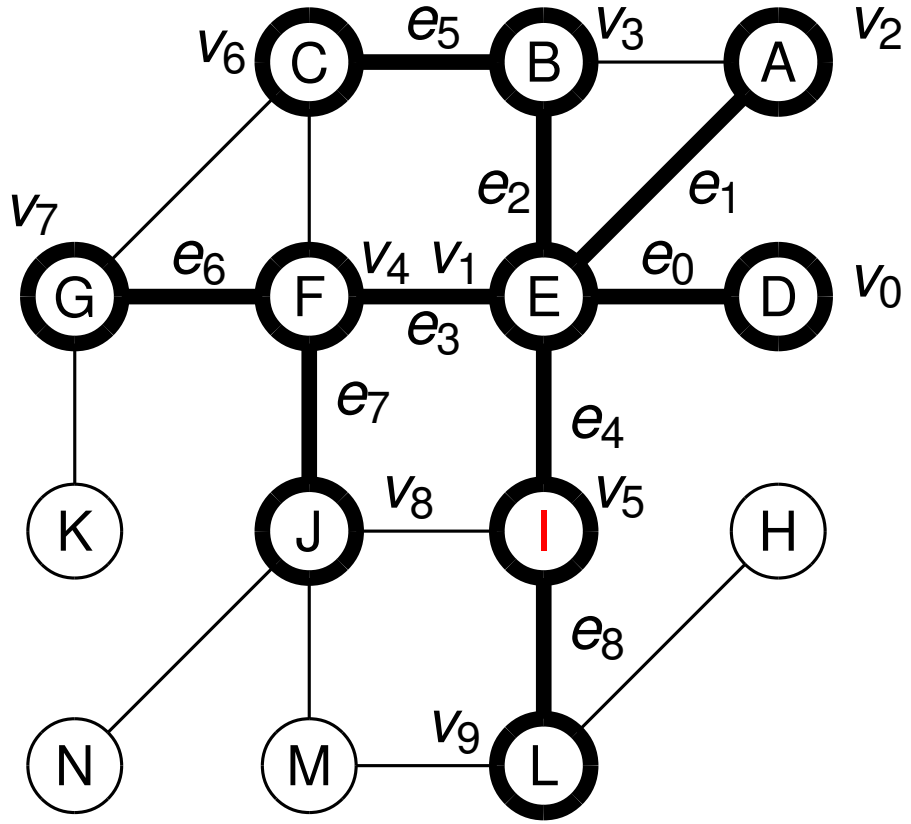
$m = 4$ Centro: $w = v_4 = F$ $n = 8$ T_8 : (árbol dibujado)

Ejemplo (Paso 4)



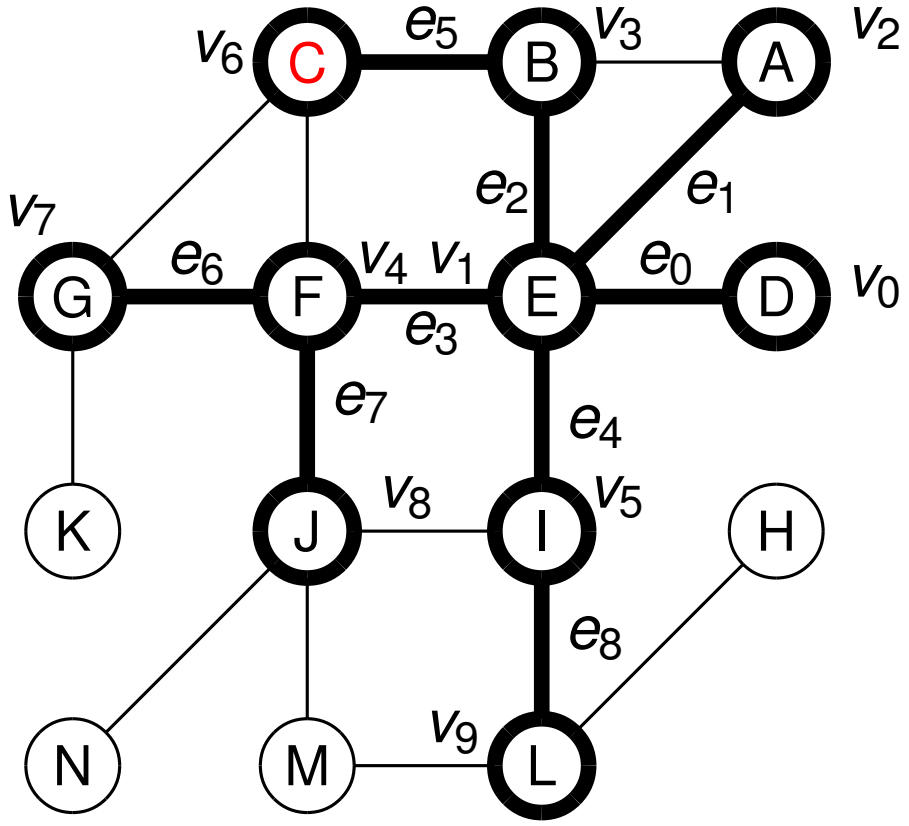
$m = 5$ Centro: $w = v_5 = I$ $n = 8$ T_8 : (árbol dibujado)

Ejemplo (Paso 3)



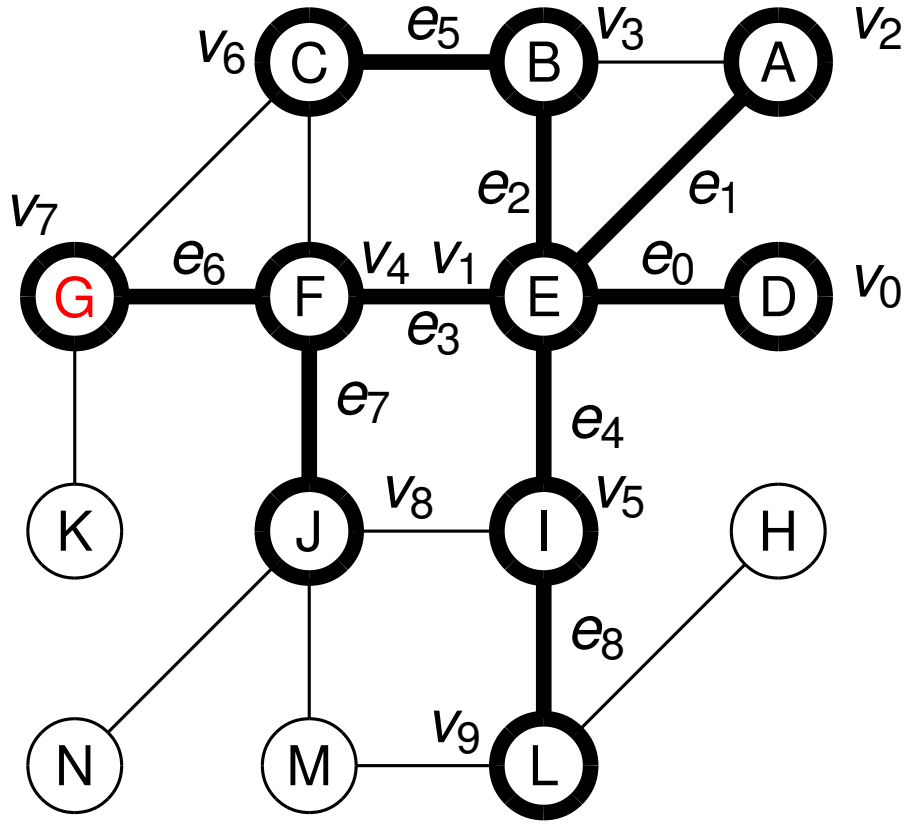
$m = 5$ Centro: $w = v_5 = I$ $n = 9$ T_9 : (árbol dibujado)

Ejemplo (Paso 4)



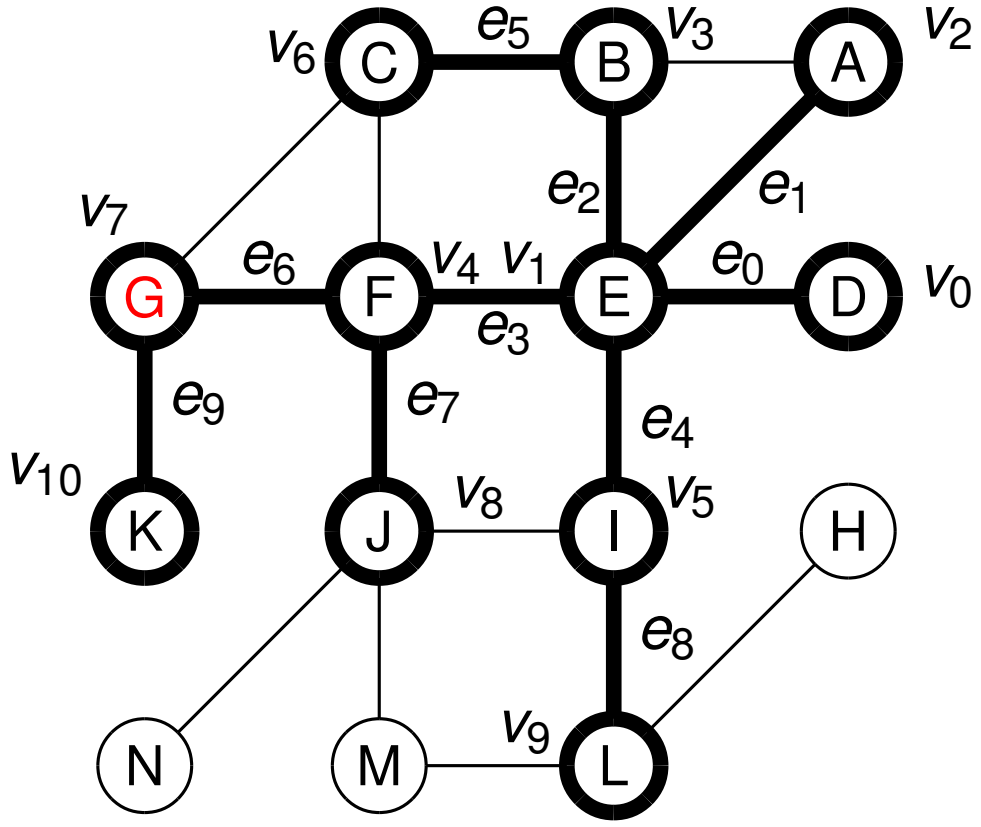
$m = 6$ Centro: $w = v_6 = C$ $n = 9$ T_9 : (árbol dibujado)

Ejemplo (Paso 4)



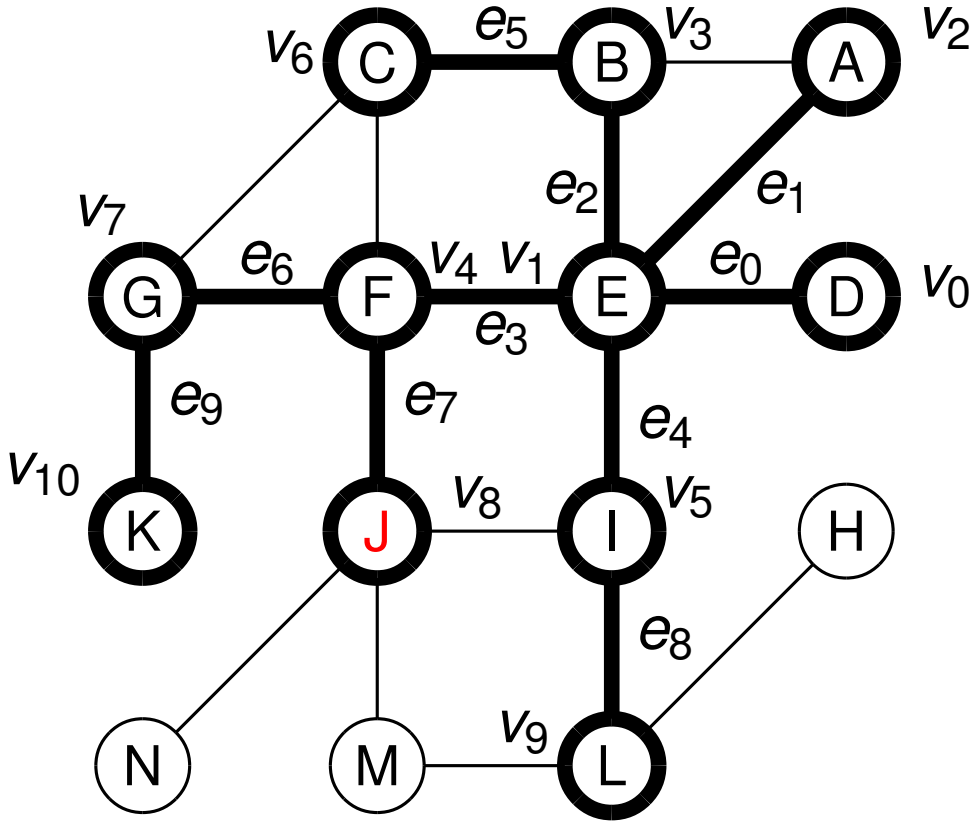
$m = 7$ Centro: $w = v_7 = G$ $n = 9$ T_9 : (árbol dibujado)

Ejemplo (Paso 3)



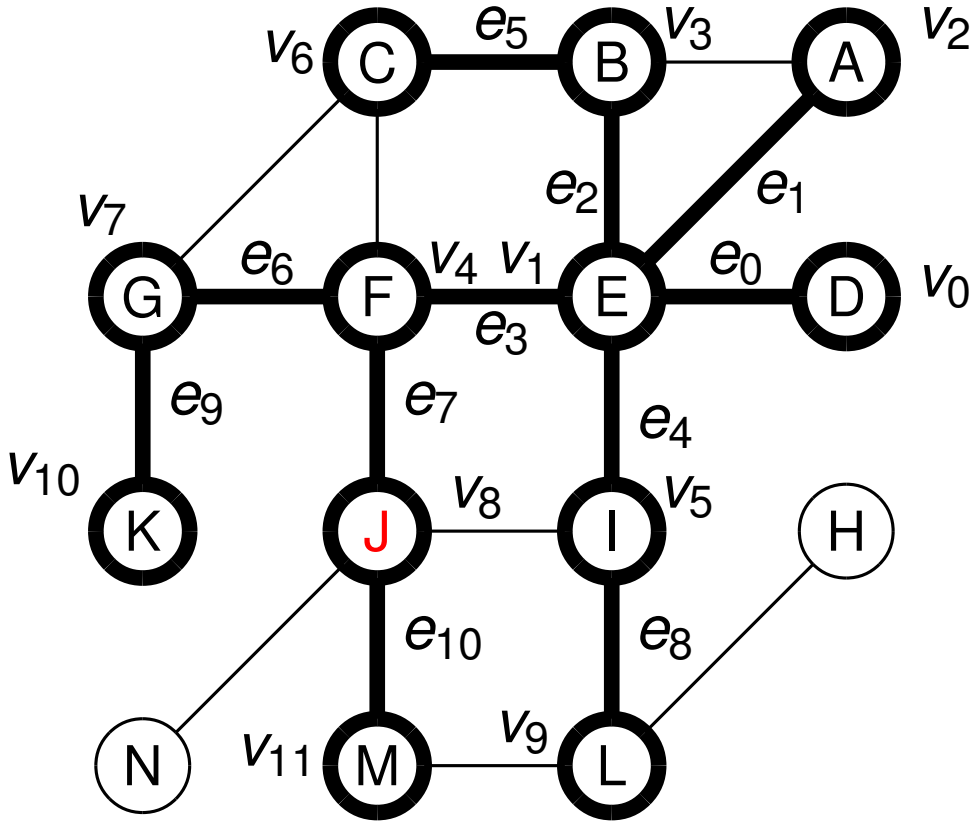
$m = 7$ Centro: $w = v_7 = G$ $n = 10$ T_{10} : (árbol dibujado)

Ejemplo (Paso 4)



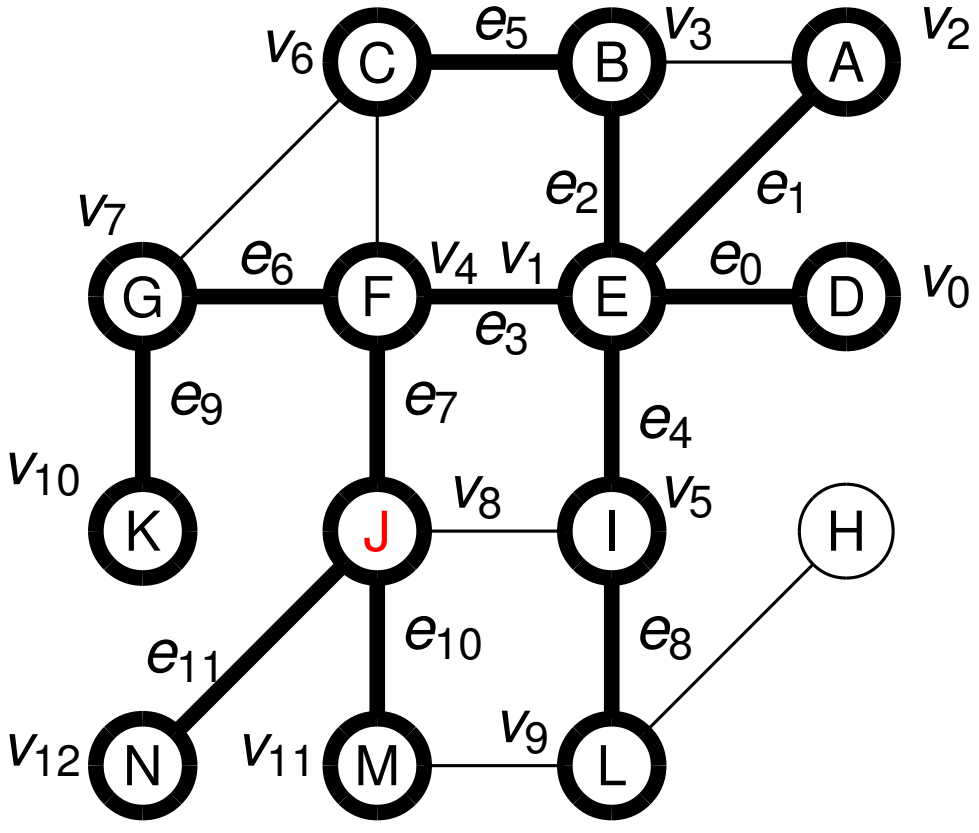
$m = 8$ Centro: $w = v_8 = J$ $n = 10$ T_{10} : (árbol dibujado)

Ejemplo (Paso 3)



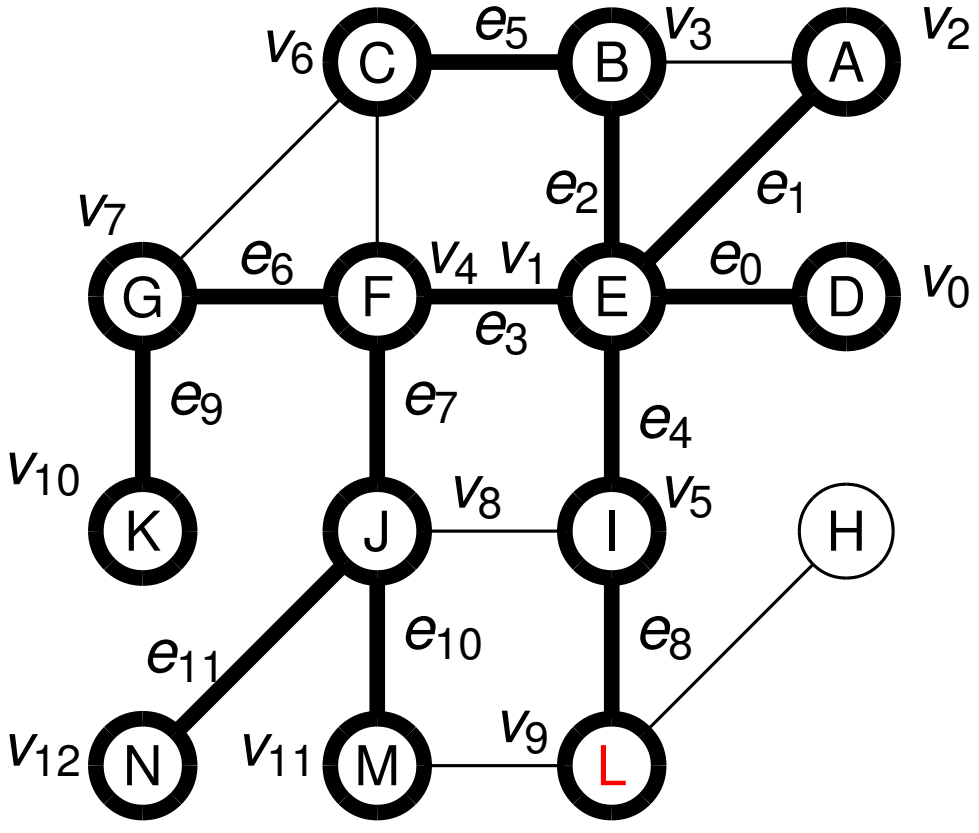
$m = 8$ Centro: $w = v_8 = J$ $n = 11$ T_{11} : (árbol dibujado)

Ejemplo (Paso 3)



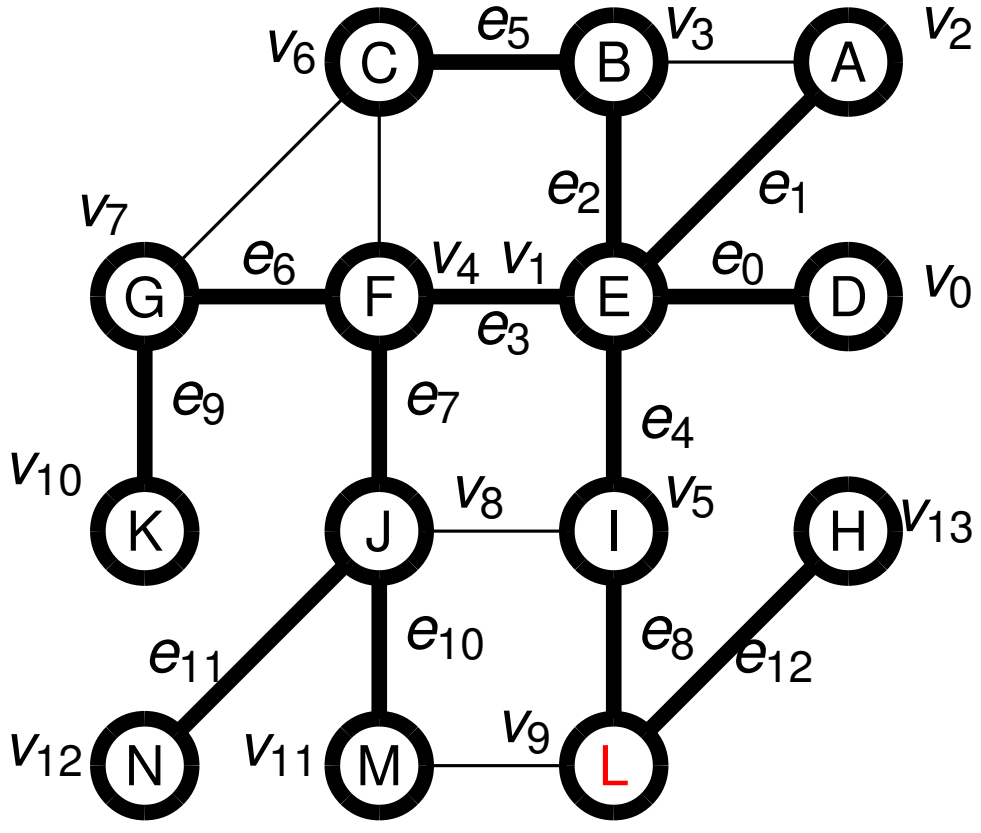
$m = 8$ Centro: $w = v_8 = J$ $n = 12$ T_{12} : (árbol dibujado)

Ejemplo (Paso 4)



$m = 9$ Centro: $w = v_9 = L$ $n = 12$ T_{12} : (árbol dibujado)

Ejemplo (Paso 3)



$m = 9$ Centro: $w = v_9 = L$ $n = 13$ T_{13} : (árbol dibujado)

Algoritmo Depth-first search (DFS)

1. Elige un vértice v_0 del grafo y define $n = 0$, $m = 0$ y $w_m = v_0$ y $w = w_m$ (w es el "centro" actual de la búsqueda). Sea T_0 el árbol sin aristas cuyo único vértice es v_0.
2. Si existe algún vértice nuevo (nuevo significa "que no es un vértice del árbol actual") que sea adyacente a w entonces
   - elige uno de ellos, v_{n+1};
   - añade a T_n el nuevo vértice v_{n+1} así como una arista cuyos extremos sean w y v_{n+1}. Formaremos, así, el siguiente árbol T_{n+1};
   - toma $w_{m+1} = v_{n+1}$, $w = w_{m+1}$ e incrementa m y n en una unidad;
   - repite el paso 2 hasta que no haya nuevos vértices adyacentes a w.
3. Si todos los vértices han sido visitados entonces el último árbol obtenido T_{n-1} es un árbol generador de G y hemos acabado. En otro caso: toma como nuevo centro w el centro anterior, es decir, $w = w_{m-1}$, réstale una unidad a m y ve al paso 2.

Ejercicio

Aplica el algoritmo DFS para calcular un árbol generador del grafo del ejemplo anterior.

Observaciones

Aunque hemos presentado los algoritmos BFS y DFS como dos métodos de cálculo de un árbol generador de un grafo conexo, pueden también aplicarse sobre grafos no conexos:

- Si partimos de un vértice inicial v, la aplicación de los algoritmos da como resultado un árbol cuyos vértices son los vértices del grafo que están conectados con v. Dicho de otro modo,
  son los vértices de la componente conexa de v.
- Si, como resultado del algoritmo, se obtienen todos los vértices del grafo, entonces éste es conexo. En caso contrario, aplicando de nuevo el algoritmo pero empezando con un vértice no obtenido, calcularemos los vértices de otra componente conexa distinta. Repitiendo el proceso podremos calcular lo vértices de todas las componentes conexas del grafo.