Pràctiques de Matemàtica Discreta: Introducció a la teoria de grafs

Sessió 6

Algorismes de cerca

Un algorisme de cerca en un graf connex G és un procediment sistemàtic l'objectiu del qual és "visitar" tots els vèrtexs de G "viatjant" a través de les arestes. Un vèrtex pot visitar-se més d'una vegada i una aresta pot travessar-se més d'una vegada al llarg del procés de cerca.

Dit d'una altra manera, un algorisme de cerca és un procés sistemàtic per a construir un subgraf de G que conté a tots els seus vèrtexs, és a dir, un subgraf generador de G.

Veurem dos algorismes per a construir un tipus particular de subgraf generador: un arbre generador. Són els següents:

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

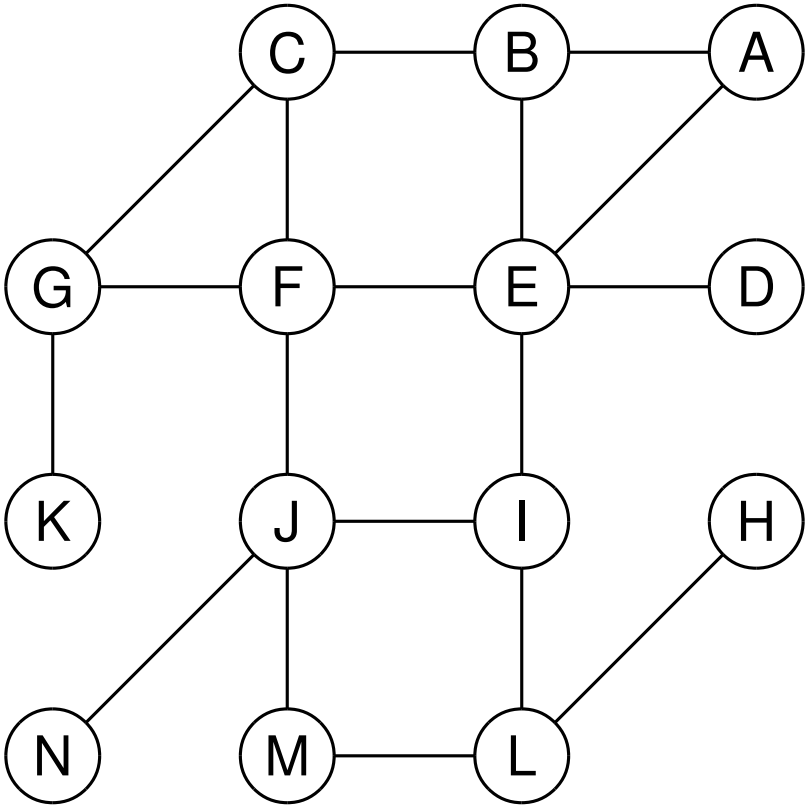
Breadth: amplada; depth: profunditat

Algorisme Breadth-first search (BFS)

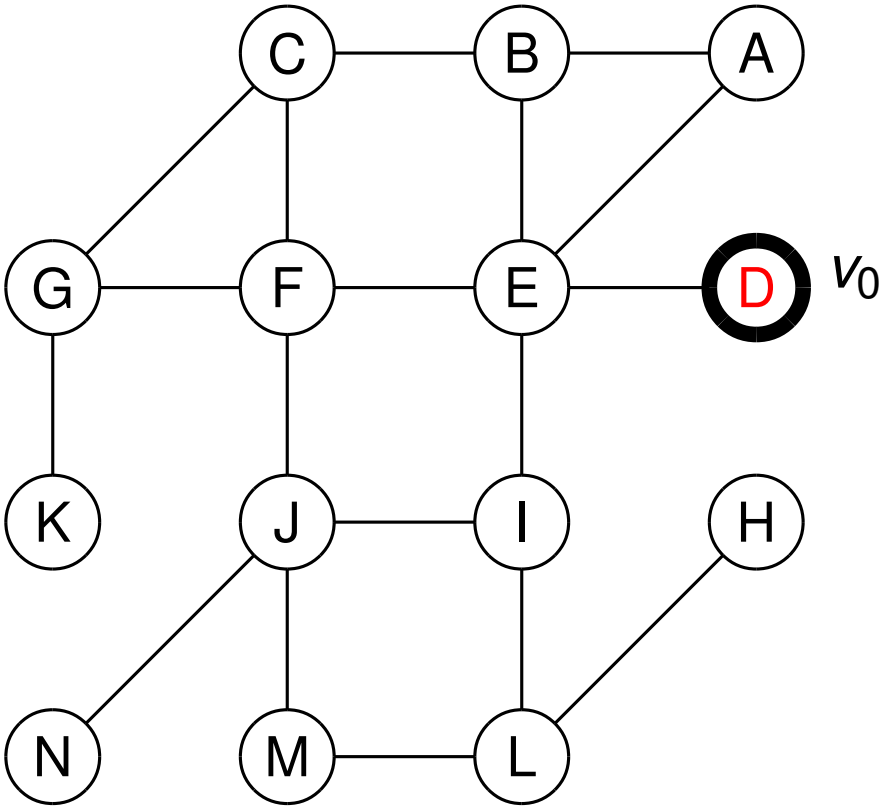
Siga G un graf connex.

1. Tria un vèrtex v_0 del graf.
2. Siguen $m = 0$, $w := v_m = v_0$ i $n = 0$. (Al vèrtex $w = v_m$ l'anomenarem "centre actual" de la cerca). Siga T_0 l'arbre sense arestes l'únic vèrtex del qual és v_0.
3. Si existeix algun vèrtex nou (nou vol dir "que no és un vèrtex de l'arbre actual") que siga adjacent a w llavors
   - tria un d'ells, v_{n+1};
   - afegeig a T_n el nou vèrtex v_{n+1} així com una aresta e_n els extrems de la qual siguen w i v_{n+1}. Formarem, així, el següent arbre T_{n+1};
   - incrementa n en una unitat;
   - repeteix el pas 3 fins que no hi haja nous vèrtexs adjacents a w.
4. Si tots els vèrtexs han sigut visitats llavors T_n és un arbre generador de G i hem acabat. En un altre cas incrementa m en una unitat, posa $w = v_m$ i torna al pas 3.

Exemple

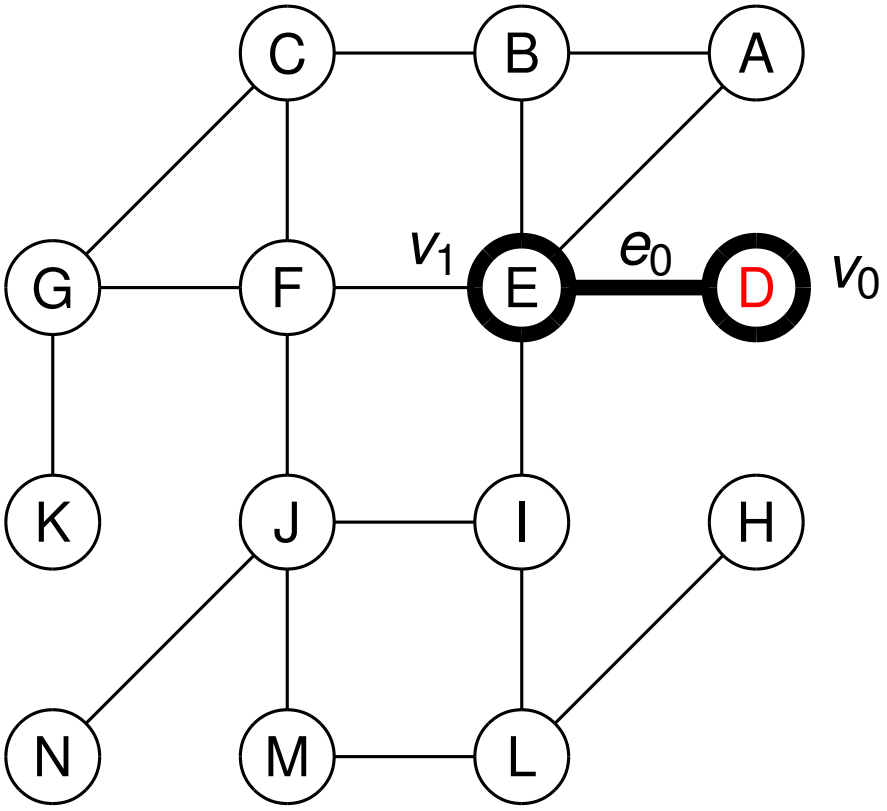


Exemple (Passos 1 i 2)



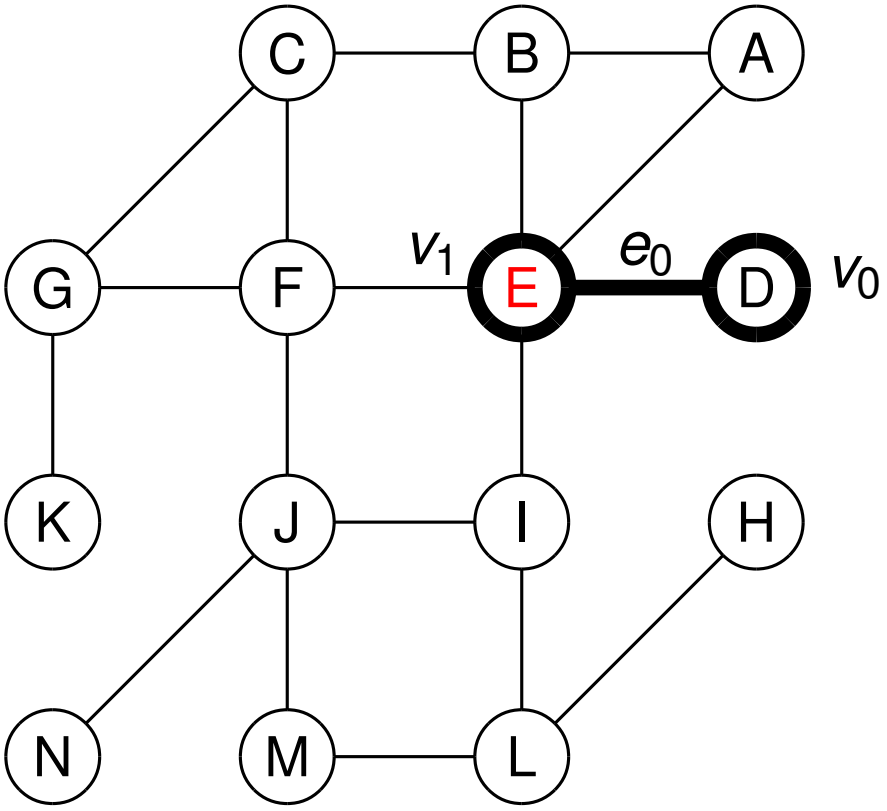
$m = 0$ Centre: $w = v_0 = D$ $n = 0$ T_0 : (arbre dibuixat)

Exemple (Pas 3)



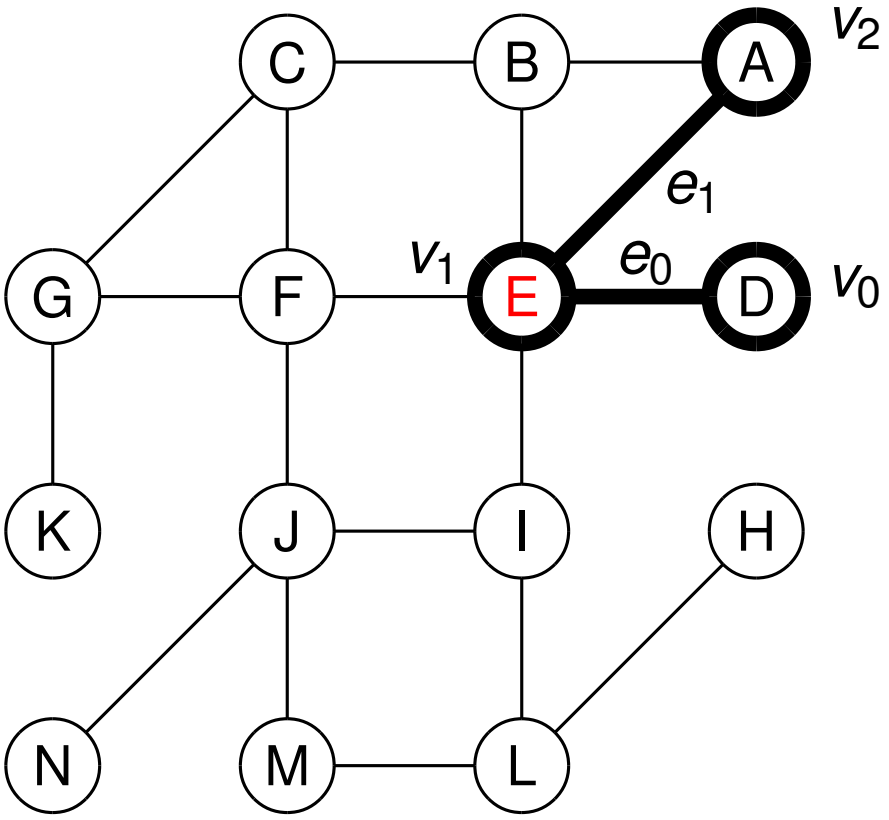
$m = 0$ Centre: $w = v_0 = D$ $n = 1$ T_1 : (arbre dibuixat)

Exemple (Pas 4)



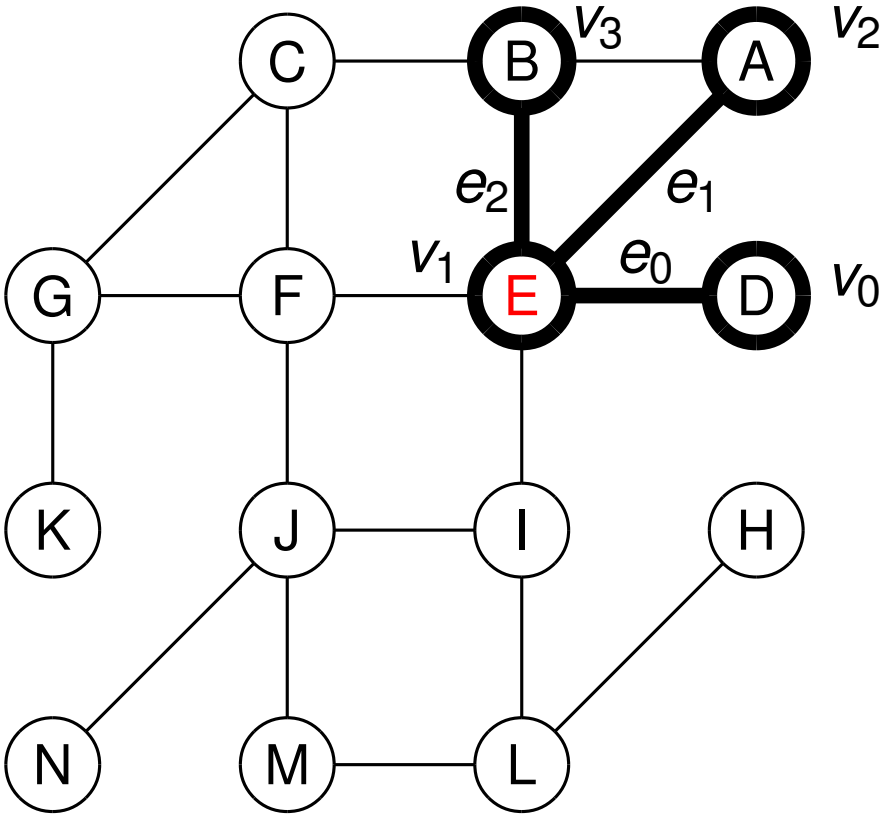
$m = 1$ Centre: $w = v_1 = E$ $n = 1$ T_1 : (arbre dibuixat)

Exemple (Pas 3)



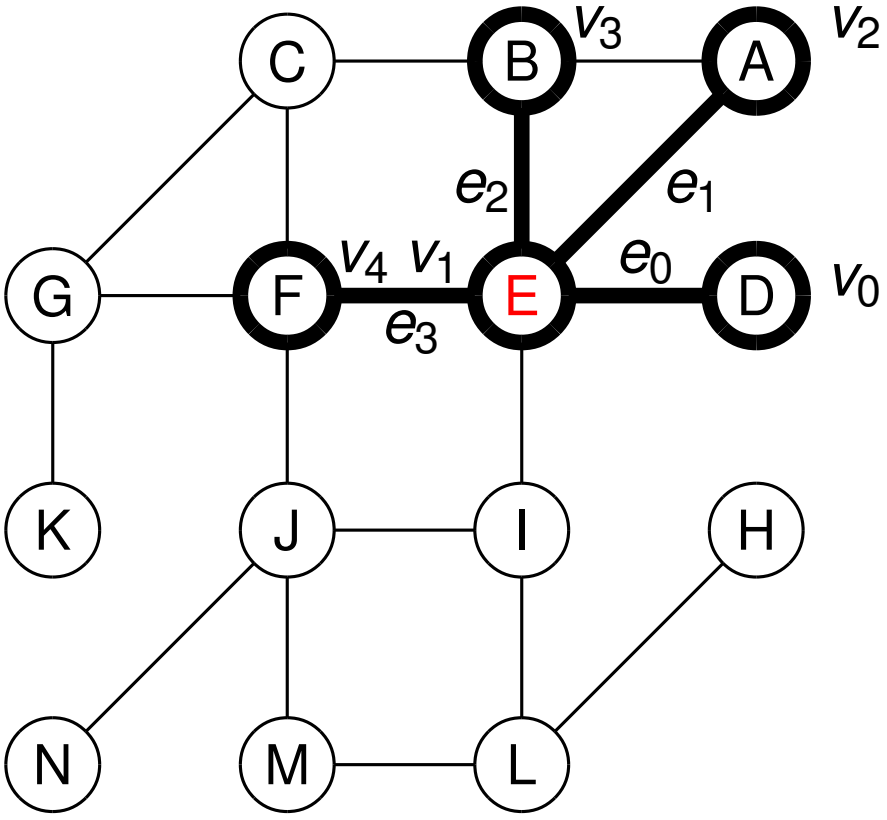
$m = 1$ Centre: $w = v_1 = E$ $n = 2$ T_2 : (arbre dibuixat)

Exemple (Pas 3)



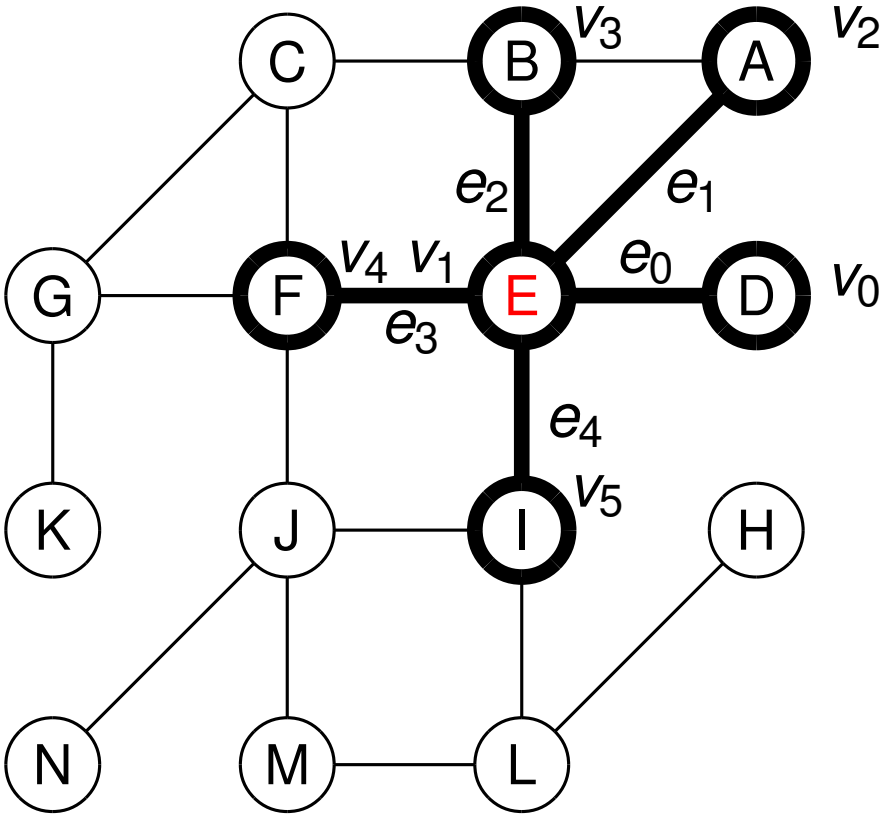
$m = 1$ Centre: $w = v_1 = E$ $n = 3$ T_3 : (arbre dibuixat)

Exemple (Pas 3)



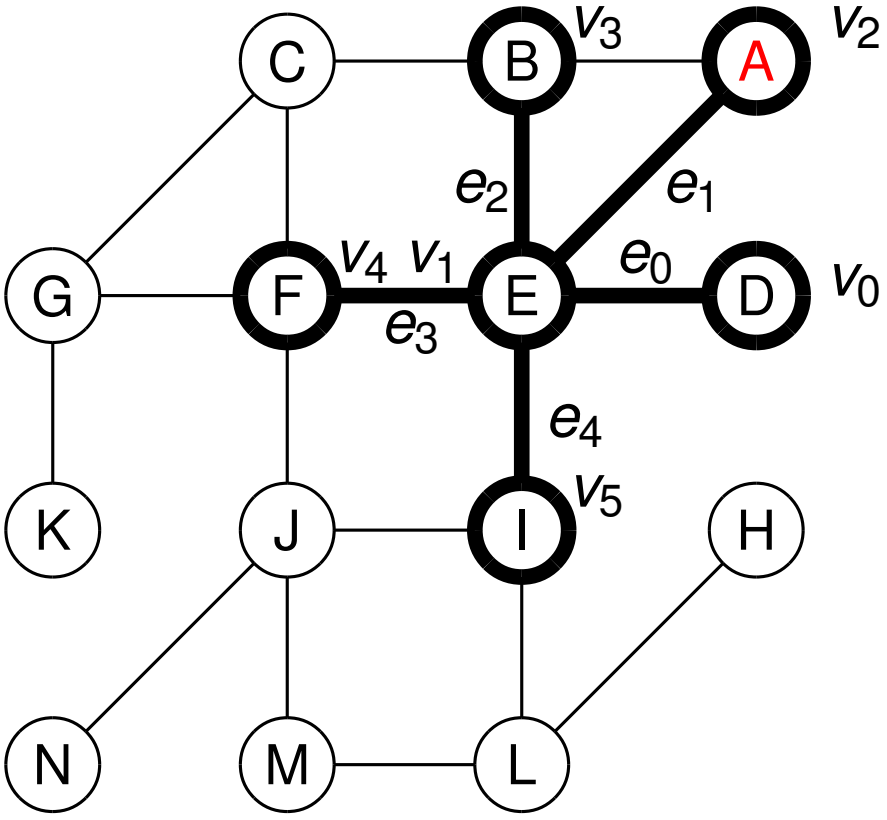
$m = 1$ Centre: $w = v_1 = E$ $n = 4$ T_4 : (arbre dibuixat)

Exemple (Pas 3)



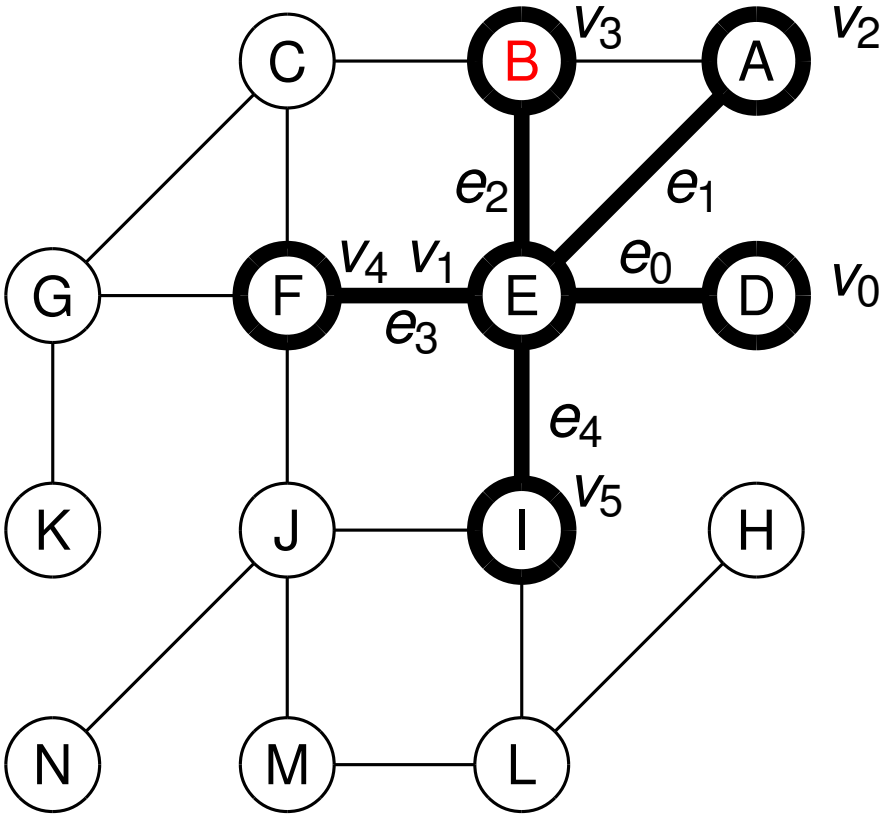
$m = 1$ Centre: $w = v_1 = E$ $n = 5$ T_5 : (arbre dibuixat)

Exemple (Pas 4)



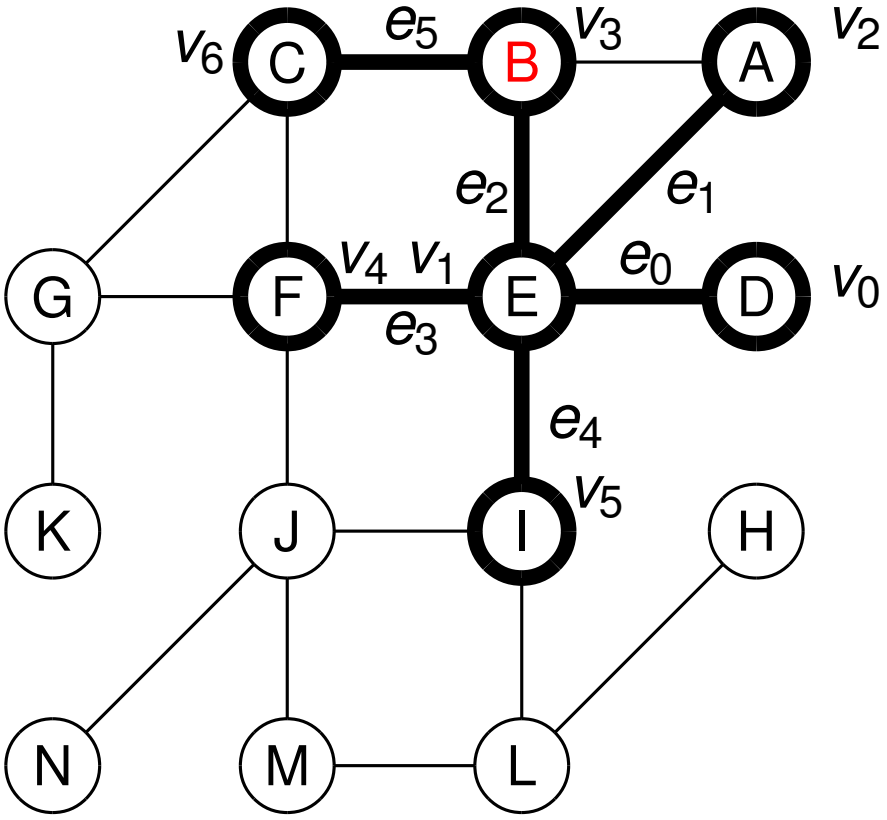
$m = 2$ Centre: $w = v_2 = A$ $n = 5$ T_5 : (arbre dibuixat)

Exemple (Pas 4)



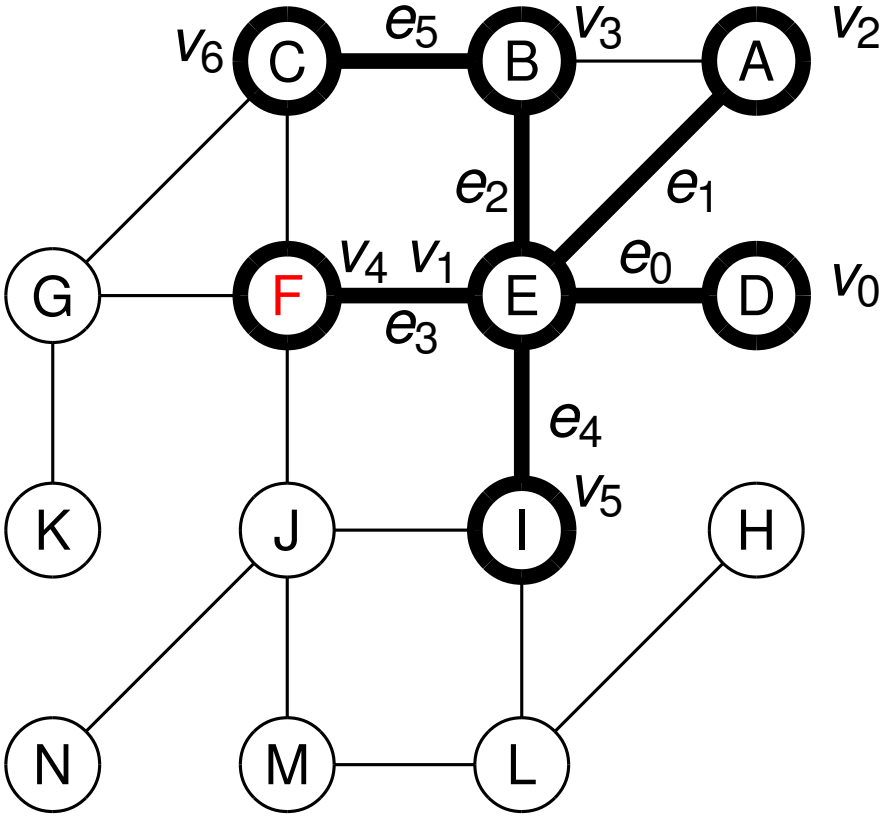
$m = 3$ Centre: $w = v_3 = B$ $n = 5$ T_5 : (arbre dibuixat)

Exemple (Pas 3)



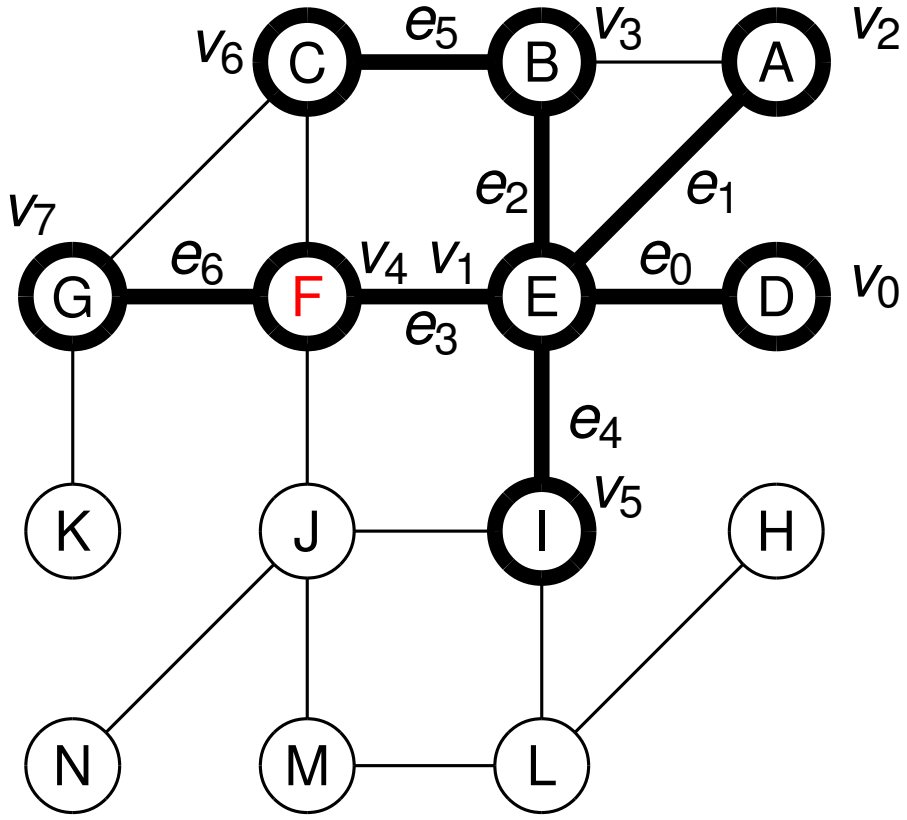
$m = 3$ Centre: $w = v_3 = B$ $n = 6$ T_6 : (arbre dibuixat)

Exemple (Pas 4)



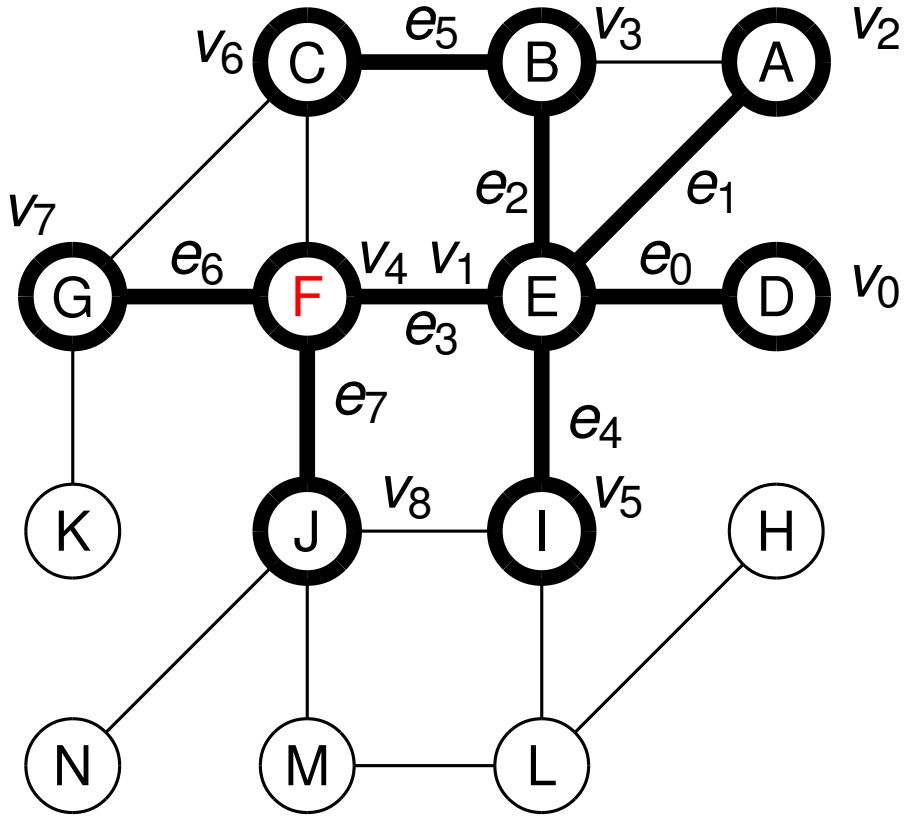
$m = 4$ Centre: $w = v_4 = F$ $n = 6$ T_6 : (arbre dibuixat)

Exemple (Pas 3)



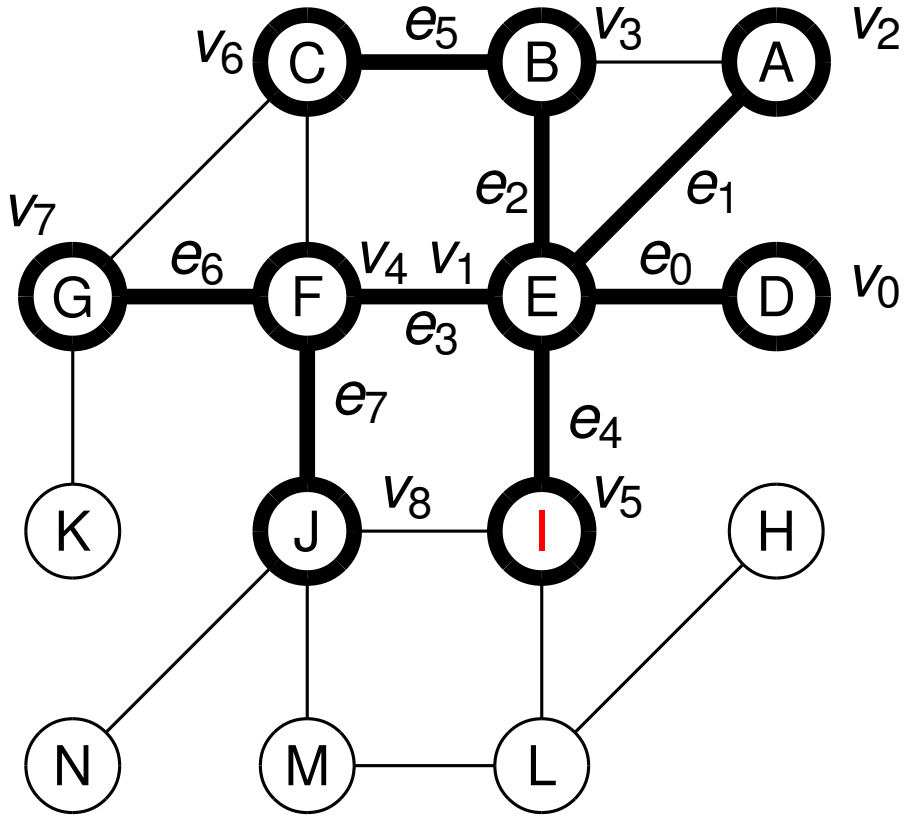
$m = 4$ Centre: $w = v_4 = F$ $n = 7$ T_7 : (arbre dibuixat)

Exemple (Pas 3)



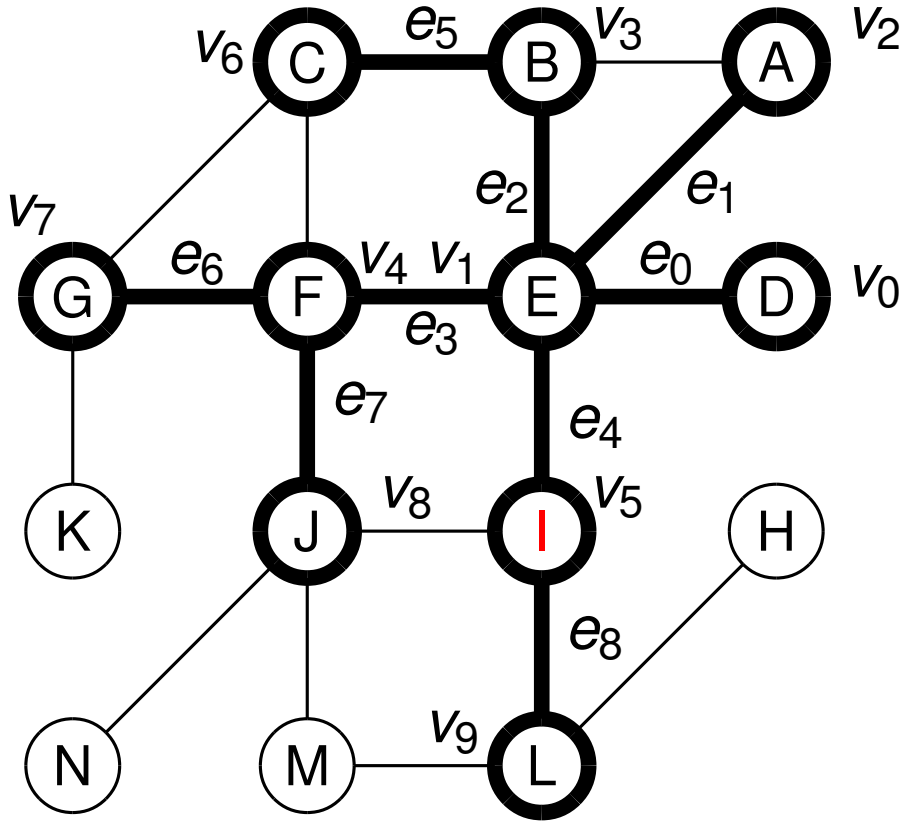
$m = 4$ Centre: $w = v_4 = F$ $n = 8$ T_8 : (arbre dibuixat)

Exemple (Pas 4)



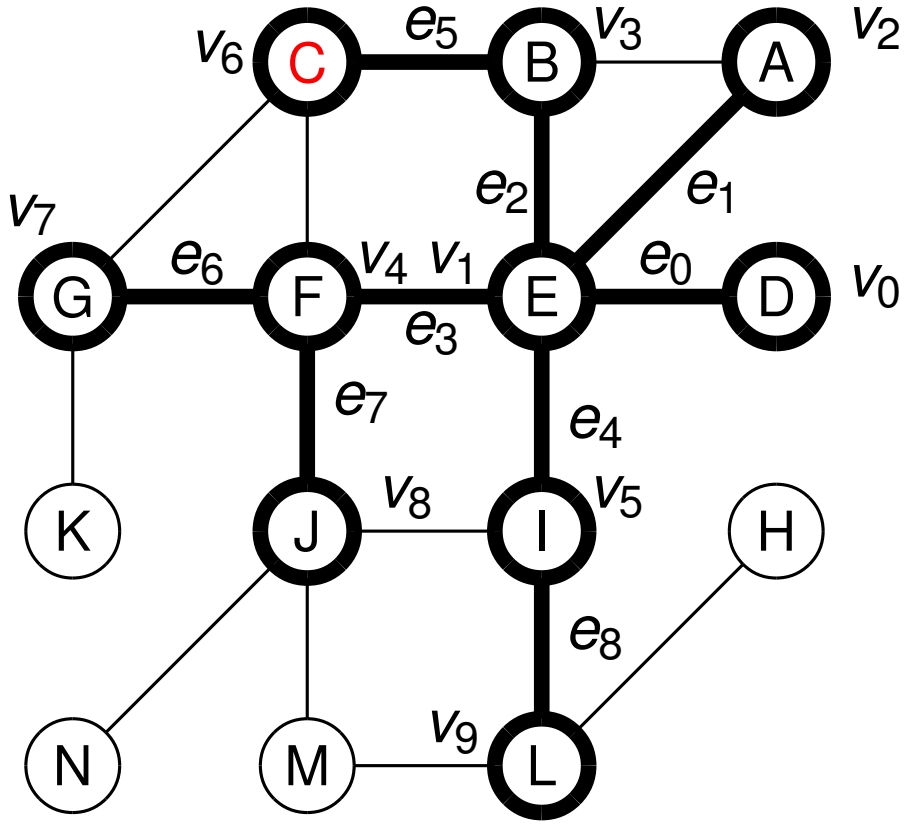
$m = 5$ Centre: $w = v_5 = I$ $n = 8$ T_8 : (arbre dibuixat)

Exemple (Pas 3)



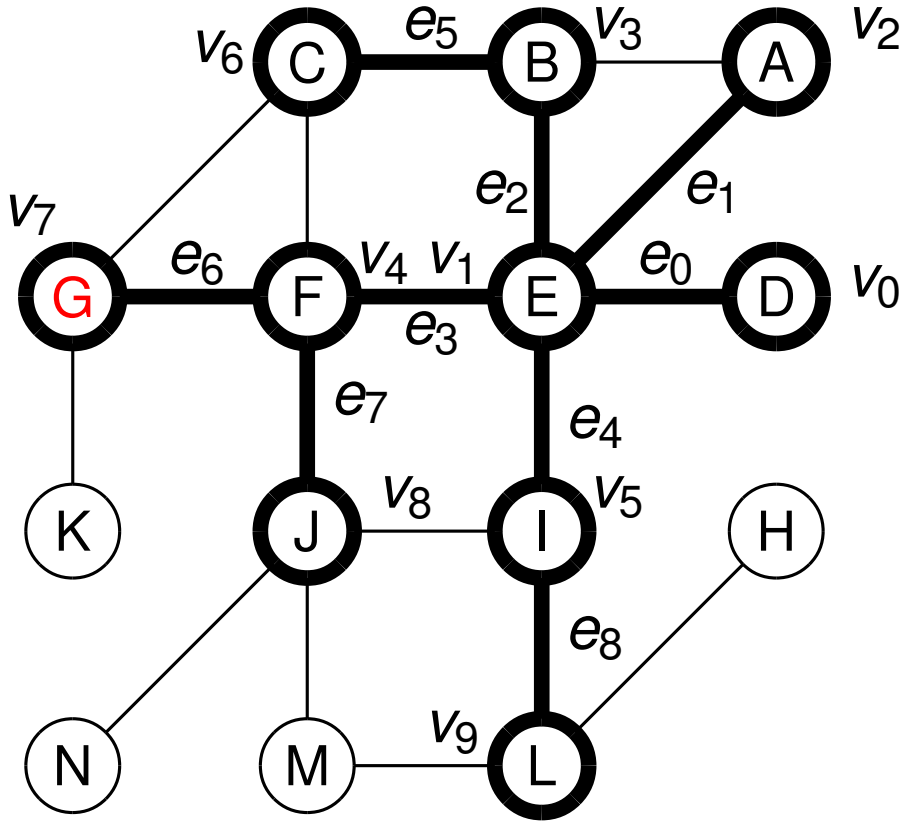
$m = 5$ Centre: $w = v_5 = I$ $n = 9$ T_9 : (arbre dibuixat)

Exemple (Pas 4)



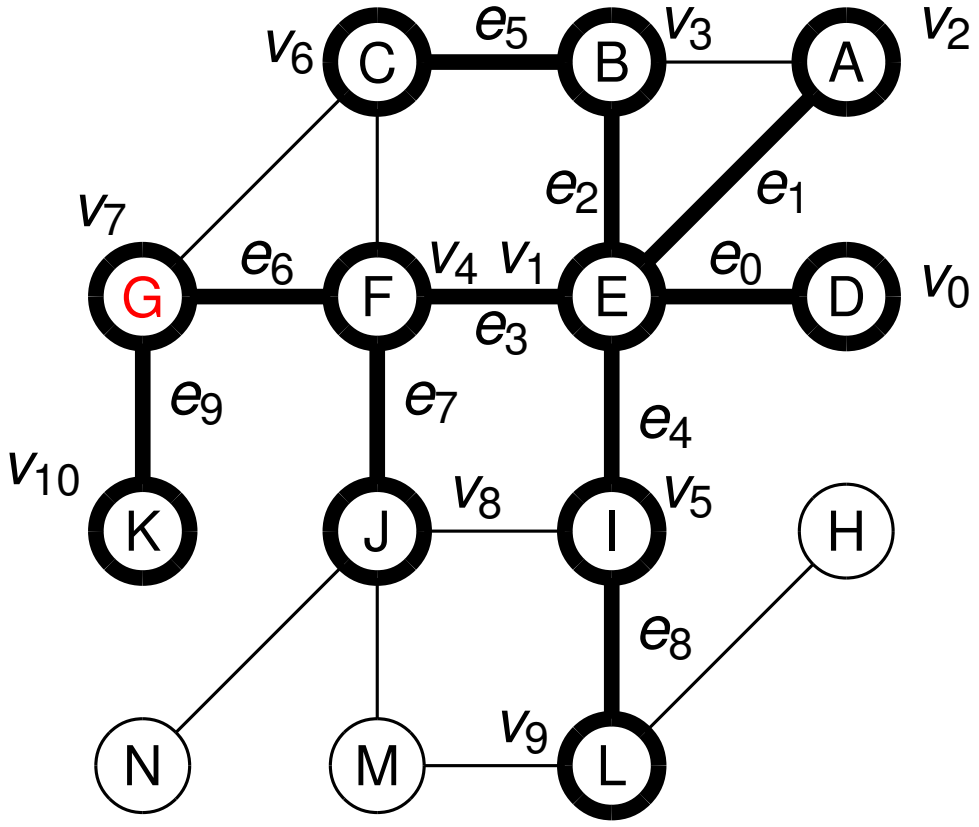
$m = 6$ Centre: $w = v_6 = C$ $n = 9$ T_9 : (arbre dibuixat)

Exemple (Pas 4)



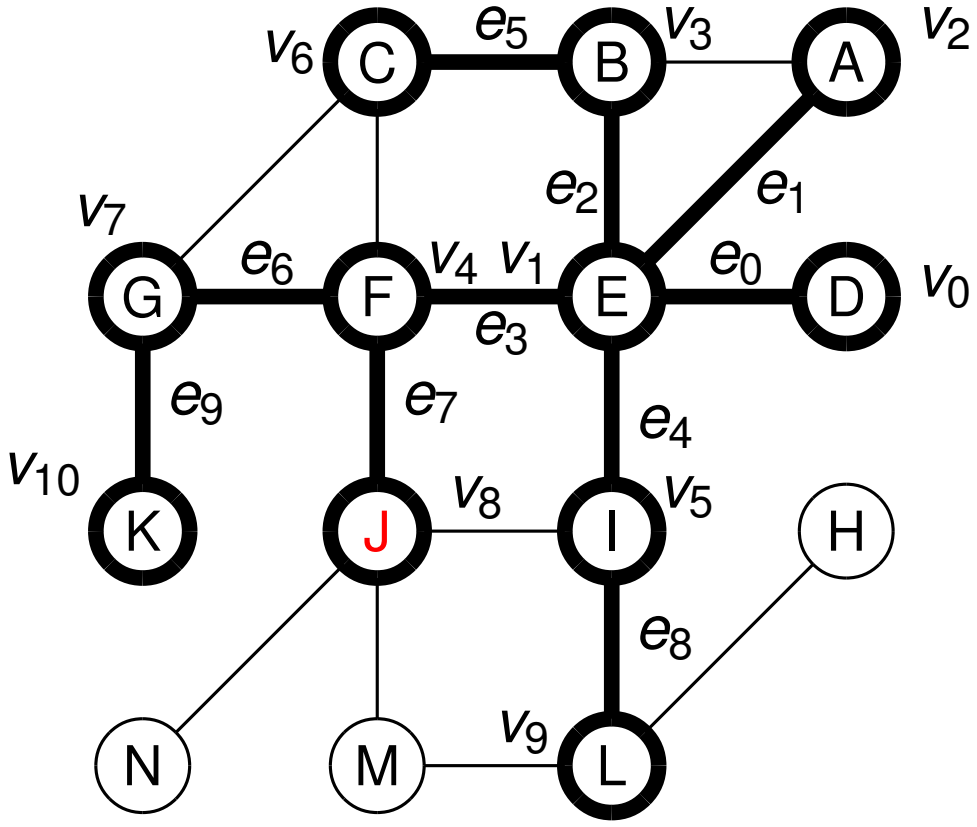
$m = 7$ Centre: $w = v_7 = G$ $n = 9$ T_9 : (arbre dibuixat)

Exemple (Pas 3)



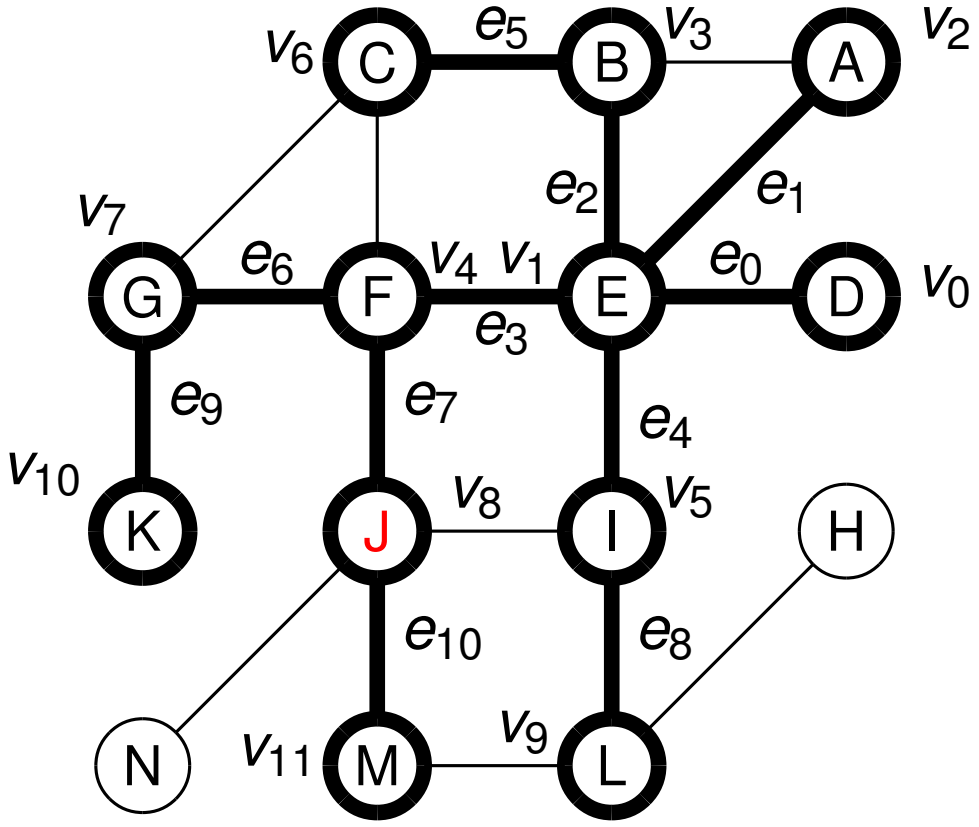
$m = 7$ Centre: $w = v_7 = G$ $n = 10$ T_{10} : (arbre dibuixat)

Exemple (Pas 4)



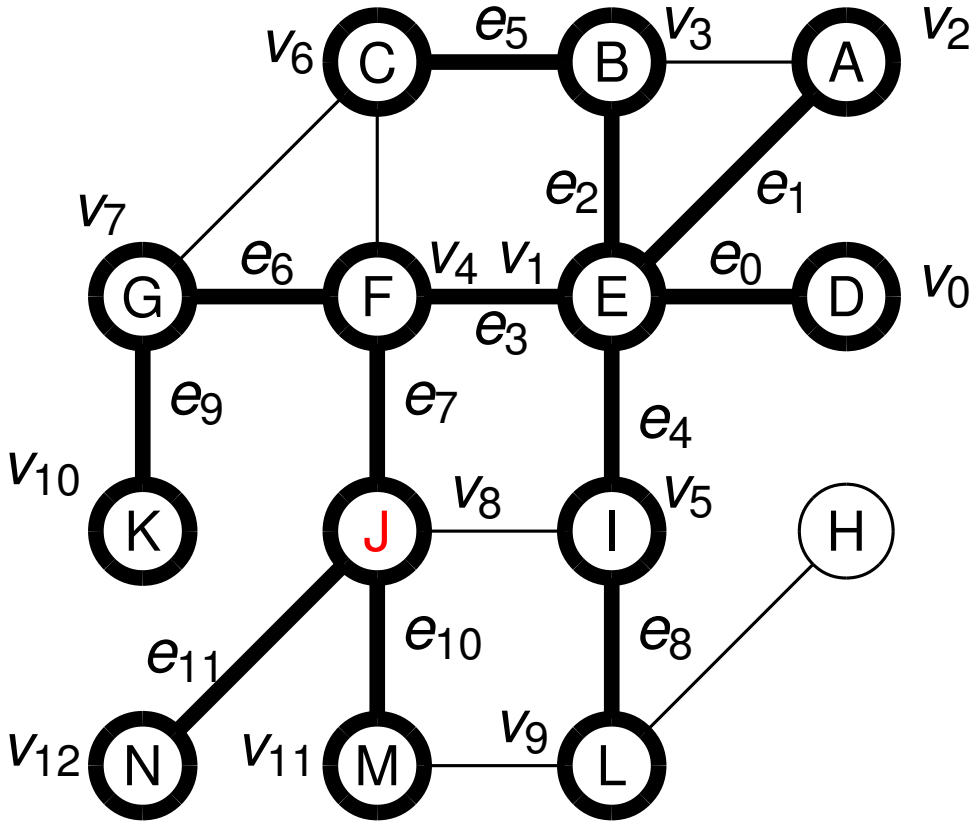
$m = 8$ Centre: $w = v_8 = J$ $n = 10$ T_{10} : (arbre dibuixat)

Exemple (Pas 3)



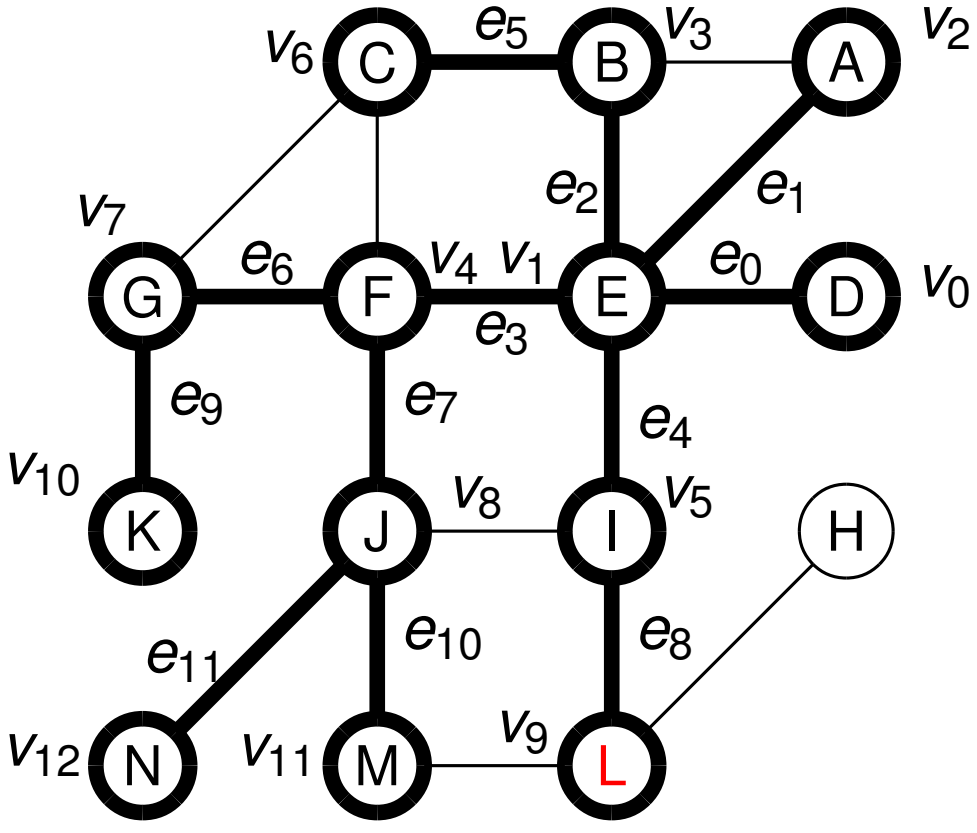
$m = 8$ Centre: $w = v_8 = J$ $n = 11$ T_{11} : (arbre dibuixat)

Exemple (Pas 3)



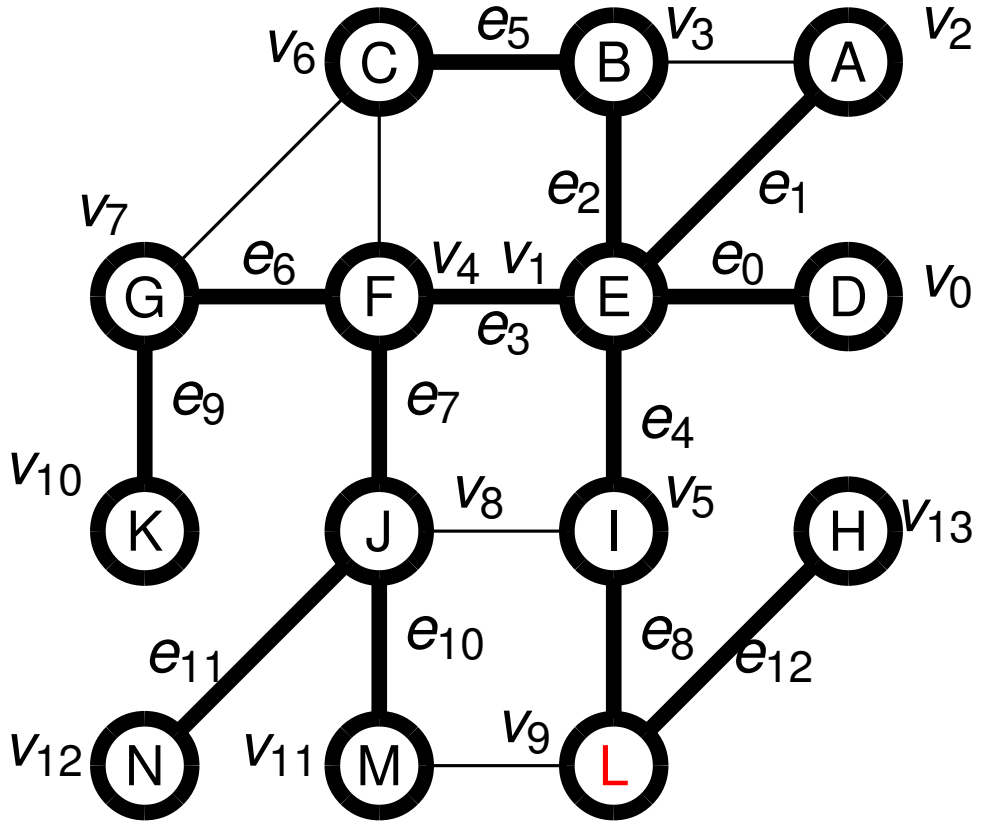
$m = 8$ Centre: $w = v_8 = J$ $n = 12$ T_{12} : (arbre dibuixat)

Exemple (Pas 4)



$m = 9$ Centre: $w = v_9 = L$ $n = 12$ T_{12} : (arbre dibuixat)

Exemple (Pas 3)



$m = 9$ Centre: $w = v_9 = L$ $n = 13$ T_{13} : (arbre dibuixat)

Algorisme Depth-first search (DFS)

1. Tria un vèrtex v_0 del graf i defineix $n = 0$, $m = 0$, $w_m = v_0$ i $w = w_m$ (w és el "centre" actual de la cerca). Siga T_0 l'arbre sense arestes l'únic vèrtex del qual és v_0.
2. Si existeix algun vèrtex nou (nou vol dir "que no és un vèrtex de l'arbre actual") que siga adjacent a w llavors
   - tria un d'ells, v_{n+1};
   - afegeig a T_n el nou vèrtex v_{n+1} així com una aresta els extrems de la qual siguen w i v_{n+1}. Formarem, així, el següent arbre T_{n+1};
   - posa $w_{m+1} = v_{n+1}$, $w = w_{m+1}$ i incrementa m i n en una unitat;
   - repeteix el pas 2 fins que no hi haja nous vèrtexs adjacents a w.
3. Si tots els vèrtexs han sigut visitats llavors l'últim arbre obtingut és un arbre generador de G i hem acabat. En un altre cas: agafa com a nou centre w el centre anterior, és a dir, $w = w_{m-1}$, resta-li una unitat a m i torna al pas 2.

Exercici

Aplica l'algorisme DFS per a calcular un arbre generador del graf de l'exemple anterior.

Observacions

Encara que hem presentat els algorismes BFS i DFS com dos mètodes de càlcul d'un arbre generador d'un graf connex, poden també aplicar-se sobre grafs no connexos:

- Si partim d'un vèrtex inicial v, l'aplicació dels algorismes dóna com resultat un arbre els vèrtexs del qual són els vèrtexs del graf que estan connectats amb v. Dit d'una altra manera, son els vèrtexs del component connex de v.
- Si, com a resultat de l'algorisme, s'obtenen tots els vèrtexs del graf, llavors aquest és connex. En cas contrari, aplicant de nou l'algorisme però començant amb un vèrtex no obtingut, calcularem els vèrtexs d'un altre component connex diferent. Repetint el procés podrem calcular el vèrtexs de tots els components connexos del graf.