Algoritmo de Prim

1. Necesita Grafo o Digrafo, valorado
2. Conexo
3. Crea un AEM(árbol de expansión mínima) y costo mínimo
4. Seleccionar un vertice inicio, marca visitado, solo se vista una vez
5. Al grafo es necesario quitar los ciclos ( punto de partida es igual al punto de llegada)
6. V(conjunto de todos los vértices)={A,B,C,D,F,G;E}

6

|  |  |  |
| --- | --- | --- |
| Grafo  V  V | Grafo Vértices | Árbol AEM |
| V  V  V  V, | V(conjunto de todos los vértices)={A,B,C,D,F,G;E}  V= {A,B,C,D,F,G,E}  W={A,D,F,B,E,C,G}  Costo= 5+6+7+7+5+9 = 39 | 7  A C  5  7  5  B  D E  9  F  G |

V

Algoritmo de Prim

|  |  |  |
| --- | --- | --- |
| Grafo | Grafo Vértices | Árbol |
| V  V  V  V  V  V  V  V  V | V={1,2,3,9,8,4,5,6,7}  W={1,2,3,9,6,7,8,4,5}  Costo= 4+9+2+4+2+1+7+10  = 39 | 7  9  2 3 4  10  4  2  4  1 9 5  8 7 6  1  2 |

Algoritmo de Kruskal

1. Necesita Grafo o Digrafo, valorado
2. Conexo
3. Crea un AEM(árbol de expansión mínima) y costo mínimo
4. Ordenar todas las conexiones de menor a mayor
5. Al grafo es necesario quitar los ciclos ( punto de partida es igual al punto de llegada)
6. V(conjunto de todos los vértices)={A,B,C,D,F,G;E}

|  |  |  |
| --- | --- | --- |
| Grafo | Grafo Vértices | Árbol AEM |
| Costo=5+5+6+7+7+9=39 | Ordenar de menor a mayor  AD 5  CE 5  DF 6  AB 7  BE 7  BC 8 No por ciclo  FE 8 No por ciclo  BD 9 No por ciclo  EG 9  FG 11 No por ciclo  DE 15 No por ciclo | Componente conexas  A C  B  D E  F  G |

|  |  |  |
| --- | --- | --- |
| Grafo | Grafo Vértices | Árbol |
| Costo=1+2+2+4+4+7+9+10=39 | PP PLL Peso  8 7 1  7 6 2  9 3 2  1 2 4  3 6 4  9 7 6 No por ciclo  8 9 7 No por ciclo  3 4 7  1 8 9  2 3 9 No por ciclo  4 5 10  6 5 11 No por ciclo  2 8 11 No por ciclo  4 6 15 No por ciclo | 2 3 4  1 9 5  8 7 6 |

PP = punto partida

PLL = punto llegada