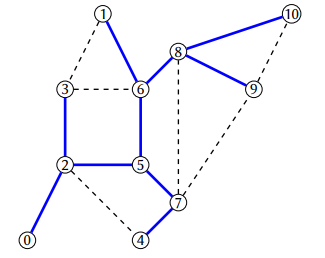
Arbre couvrant (spanning tree)

dans un graphe G à V sommets

est un sous-graphe

* connexe et
* acyclique de G
* contenant
  + tous les sommets de G
  + V − 1 arêtes.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algorithm |  |  | Code | Complexité |
| Type | Nom |  | Java | O |
| Glouton | Kruskal | • On considère les arêtes une à une, dans l’ordre croissant de leurs  poids.  • Chacune des arêtes est choisie si elle ne forme pas de cycle avec les  arêtes déjà choisies précédemment.  • Dans le cas contraire, elle est ignorée, et on procède avec l’arête  suivante dans la liste.  • Lorsque toutes les arêtes ont été examinées, les arêtes choisies  forment l’arbre couvrant minimum |  |  |
|  | Prim | 1. Initialiser l’arbre actuel T au seul sommet initial 0,  2. ajouter à T l’arête de poids minimum parmi toutes les arêtes ayant  exactement une extrémité dans T,  3. répéter l’opération précédente jusqu’à ce que T contienne V − 1  arêtes. | public PrimMST(EdgeWeightedGraph G)  {  edgeTo = new Edge[G.V()];  distTo = new double[G.V()];  marked = new boolean[G.V()];  for (int v = 0; v < G.V(); v++)  distTo[v] = Double.POSITIVE\_INFINITY;  pq = new IndexMinPQ<Double>(G.V());  distTo[0] = 0.0;  pq.insert(0, 0.0); // Initialize pq with  0, weight 0.  while (!pq.isEmpty())  visit(G, pq.delMin()); // Add closest  vertex to tree.  } |  |
|  | Dijkstra |  |  |  |