


TD4: Compléments et autres algorithmes

Bornes inférieures de complexité

Exercice 1.

Exo TD : Borninf


L'enveloppe convexe d'un ensemble de n points $\mathcal{P} = \{P_1, \dots, P_n\}$ du plan est le plus petit polygone convexe C contenant tous ces points. Il est facile (?) de voir que les sommets de C sont des points de \mathcal{P} . On codera C par la liste de ses sommets donnée dans le sens direct en partant du sommet de C d'abscisse minimale.

-  Montrer que tout algorithme permettant de calculer l'enveloppe convexe des n points \mathcal{P} a une complexité en $\Omega(n \log n)$. On pourra considérer n nombres positifs x_1, \dots, x_n et lancer un calcul d'enveloppe convexe sur les points $P_1 = (x_1, x_1^2)$, $P_2 = (x_2, x_2^2), \dots, P_n = (x_n, x_n^2)$.

Arbre couvrant de poids minimum


Exercice 2.

Exo TD : Voirie

-  Modéliser le problème de la ville embourbée du cours par un graphe valué et le résoudre 'à la main' en utilisant l'algorithme de Kruskal.

Exercice 3.

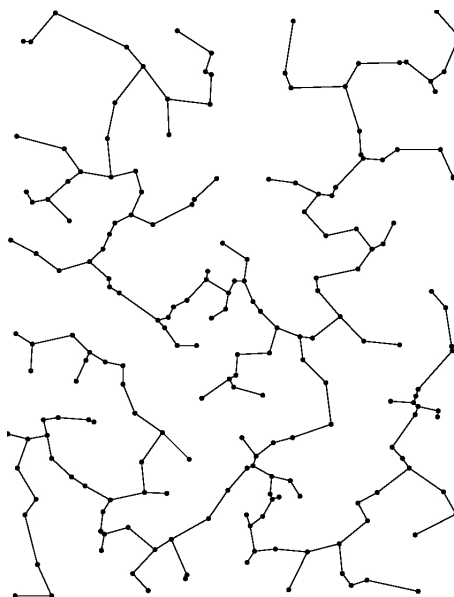
Exo TP : Kruskal

-  Implémenter l'algorithme de Kruskal donné dans le cours. On codera un graphe par un nombre de sommet et une liste d'arêtes. Les sommets seront représentés par les entiers $\{0, \dots, n-1\}$. Pour chaque arête ij du graphe, la liste d'arête devra contenir le triple $[i, j, p(ij)]$. Tester votre algorithme sur l'exemple de la question précédente.

Exercice 4.

Exo TP : L'arbre euclidien minimum


- Générer un ensemble de n points P_1, \dots, P_n dans le plan (voir Td/Tp 2).
- Créer le graphe valué G dont les sommets sont P_1, \dots, P_n et qui contient toutes les arêtes possibles, c'est-à-dire, tous les $P_i P_j$ pour $i \neq j$. Chaque arête $P_i P_j$ aura pour valuation le carré de la distance euclidienne entre P_i et P_j (on prend le carré pour garder des nombres entiers, mais ce n'est pas obligatoire...)
- Appliquer l'algorithme de Kruskal à votre graphe G . L'APCM obtenu s'appelle l'arbre euclidien minimum :



k plus proches voisins

Exercice 5.

Exo TP : *Bon voisinage*

-  Implémenter l'algorithme de calcul des k plus proches voisins. On l'appliquera aux problèmes de classification et de clusterisation. Des exemples, données, etc sont accessible ici :
- https://pixees.fr/informatiquelycee/n_site/nsi_prem_knn.html
<https://diu-eil.univ-lyon1.fr/bloc2/1.3-kNN-TP.pdf>

Tour de hanoï

Exercice 6.

Exo TD/TP : *Viet Nam*

1. Prouver proprement le théorème de l'analyse de l'algorithme HANOI.
2. Implémenter l'algorithme récursif de résolution du problème de la tour de Hanoï. Essayer d'avoir un rendu graphique si possible.