

1 Analyse asymptotique théorique

1.1 Algorithme de glouton «

En analysant notre algorithme de glouton nous pouvons écrire :

- Opérations élémentaires de calcul de distance $O(1)$ « dist ».
- La matrice de cout de distance à un temps de calcul de $O(n^2)$ « cost_matrix ».
- Le calcul de nœud de l'index du nœud le plus proche de $O(n)$ « find_nearest ».
- La fonction qui construit la solution et calcul la distance trouvée $O(n^2)$ « tsp_greedy ».

Au total, la complexité du temps de calcul globale est $O(n^2)$ (règle du max).

NB : Nous avons cherché dans les littératures sur la complexité de l'algorithme de glouton pour le problème de voyageur de commerce TSP, nous avons trouvé que la complexité asymptotique est de l'ordre $O(n^2 \log n)$. Donc soit notre analyse ci-dessous est fausse, soit notre algorithme est erroné.

1.2 Programmation dynamique

L'algorithme de programmation dynamique se compose des étapes suivantes, l'une à la suite de l'autre:

- Opérations élémentaires de calcul de distance $O(1)$ « dist ».
- La matrice de distance à un temps de calcul de $O(n^2)$ « build_cost_matrix ».
- Le nombre de sous-ensembles possibles peut être au maximum $O(2^n)$ mémoire. Chaque sous-ensemble peut être résolu en plusieurs $O(n)$ temps. Ce qui donne une complexité de l'ordre $O(n^2 2^n)$ « get_minimum ».

Au total, la complexité de du temps de calcul globale est $O(n^2 2^n)$ (règle du max).

1.2.1 Algorithme d'approximation

L'algorithme de programmation d'approximation se compose des étapes suivantes, l'une à la suite de l'autre :

- Opérations élémentaires de calcul de distance $O(1)$ « distanceBtw ».
- La construction des nœuds se fait $O(n^2)$ « build_nodes ».
- La distance entre les nœuds se fait $O(n^2)$ « distanceOfEdges ».
- L'algorithme de MST se fait $O(n^2)$ « spanning_tree ».
- « saveDistance » et « loadDistance » se fait en fait $O(n^2)$

Au total, la complexité de du temps de calcul globale est $O(n^2)$ (règle du max).