Mini-projet
d'Algorithmique avancée

ENSSAT INFO 2

2024-2025

Présentation du problème

Soit un ensemble S de n points du plan avec $n > 2$, dont les abscisses valent 1, 2, ..., n et dont les ordonnées sont quelconques. Par abus de langage, on appelle point k le point d'abscisse k. On cherche la meilleure approximation de cet ensemble par une ligne brisée, c'est-à-dire une suite de segments de droite dont les extrémités sont des points de S. On peut représenter une telle ligne par les abscisses des points sur lesquels elle s'appuie en imposant que la première est 1 (une ligne brisée part du premier point de S) et la dernière est n (une ligne brisée doit se terminer par le dernier point de S). A titre d'exemple, on considère les quatre cas ci-après où on a quatre lignes brisées, à savoir : (1, 4, 8), (1, 5, 8), (1, 4, 7, 8) et (1, 3, 4, 8).

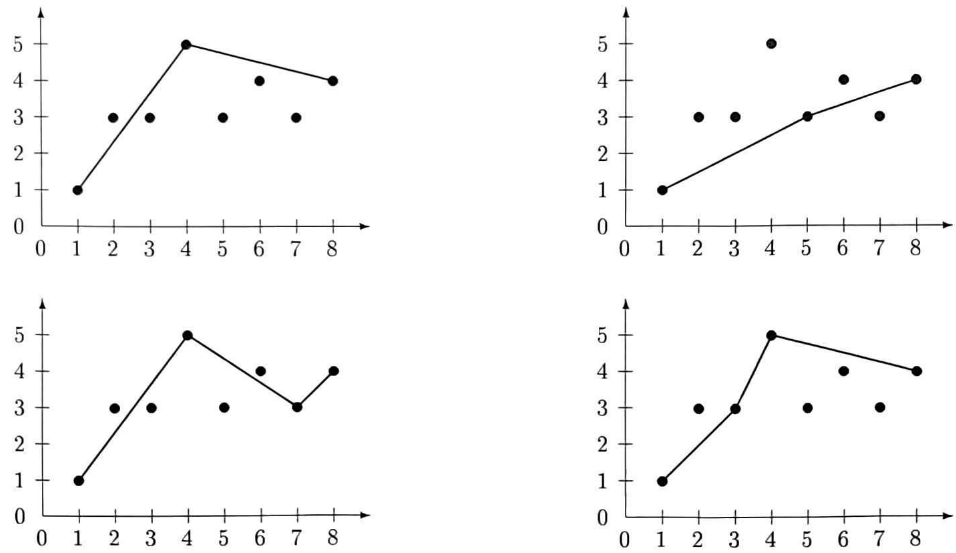


Fig. 1 : Quatre solutions possibles

La qualité d'une approximation de S par une ligne brisée se mesure pour partie en calculant la somme SD des distances euclidiennes des points de S au segment de droite dont les extrémités les encadrent. Pour la première figure, SD est la somme de la distance des points 1, 2, 3 et 4 au premier segment, soit environ $0+0.35+0.35+0 = 0.7$, et de la distance des points 4, 5, 6, 7 et 8 au second segment, soit à peu près $0+1.6+0.4+1.2+0 = 3.2$. On ajoute à SD un terme positif $m * C$ proportionnel au nombre m de segments composant la ligne brisée. Dans l'exemple des deux dernières figures, en fixant C à 1.5, ce terme vaut 4.5 puisque la ligne brisée est constituée de trois segments. On dit que l'approximation de S est d'autant meilleure que la somme $(SD + m * C)$ est petite.

Question préliminaire : Combien y a-t-il de lignes brisées avec un ensemble S ayant respectivement $n = 2$, 3 et 4 points (faire un schéma). Établir le nombre de lignes brisées pour n supérieur ou égal à 2.

A. Essais successifs

On envisage de calculer la meilleure approximation d'un ensemble S donné par une méthode à essais successifs de type *solution à trous*.

Question 1. Donner les éléments d'une procédure à essais successifs *appligbri* (paramètre i, vecteur X, S_i, *satisfaisant*, *enregistrer*, *soltrouvée*, *optimal*, *optencorepossible*, *défaire*) calculant la meilleure approximation d'un ensemble S de n points ($n > 2$). On précisera également les principales variables globales utilisées (et leur objet) ainsi que l'appel initial à la procédure *appligbri*. On utilisera une fonction $distance(i, j)$ qui délivre la somme des distances des points i à j au segment d'extrémités i et j.

Question 2. Analyser la complexité temporelle de cette procédure à essais successifs.

Question 3. Proposer une (ou des) condition(s) d'élagage à introduire dans *optencorepossible* et en montrer l'intérêt par des mesures expérimentales.

Question 4. On décide de calculer le coût de l'approximation particulière suivante : on joint les points 1 et 3, puis 3 et 5, et ainsi de suite. Comment peut-être utilisée cette connaissance pour construire une procédure à essais successifs de résolution du problème de recherche de la meilleure approximation ?

B. Programmation dynamique

On envisage maintenant de calculer la meilleure approximation d'un ensemble S donné par programmation dynamique.

Question 1. En appelant $SD_{i,j}$ la somme des distances euclidiennes des points $(i + 1)$ à $(j - 1)$ au segment joignant les points i et j, donner la récurrence complète définissant la valeur de l'approximation optimale notée *approx-opt*$(1, n)$.

Question 2. Préciser la structure tabulaire à utiliser et indiquer la stratégie de remplissage de cette structure.

Question 3. Indiquer les complexités temporelle et spatiale de la procédure obtenue.

C. Questions complémentaires

1. Donner votre opinion sur l'effort de conception de ces deux algorithmes.

2. Avec la méthode des essais successifs, jusqu'à quel nombre de points (n) peut-on aller en pratique (exécution en moins de 2 minutes) ?

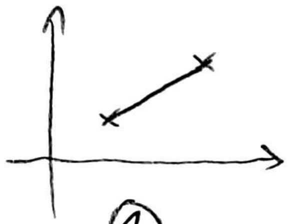
3. Serait-il possible de résoudre le problème posé à l'aide d'une solution de type "diviser pour régner" ? (justifier votre réponse)

Livrables attendus

Le mini-projet est effectué en binôme. Deux livrables par binôme sont attendus à l'issue du mini-projet :
1. Compte-rendu du mini-projet. Le compte-rendu contiendra *a minima*: une page de garde, une introduction, les réponses à toutes les questions de l'énoncé incluant les explications claires des solutions développées, les algorithmes écrits en **pseudo-langage** et **commentés**, une étude de la complexité de chaque algorithme, les jeux d'essai motivés (complets, avec étude expérimentale permettant d'exhiber empiriquement les gains découlant de la mise en œuvre des critères d'élagage choisis), une conclusion incluant un véritable bilan du projet (rappel et comparaison des solutions proposées, perspectives), les listings commentés en annexe. Le rapport hors annexes ne devra pas dépasser 10 pages.
ATTENTION : Le compte-rendu hors annexes doit pouvoir être lu indépendamment du code (contenir toutes les informations nécessaires à la compréhension et l'évaluation des solutions proposées).
2. Sources du mini-projet. Les fichiers sources commentés de votre code devront être fournis, incluant un fichier README (le fichier README expliquera comment installer et utiliser votre/vos programme(s)).

Les livrables devront être envoyés par mail à votre encadrant de TP dans un mail intitulé *"[INFO2 Algorithmique Avancée] Mini-projet du binome B_1 / B_2"* où B_1 et B_2 sont à remplacer par les noms des binômes.

n=2 points:

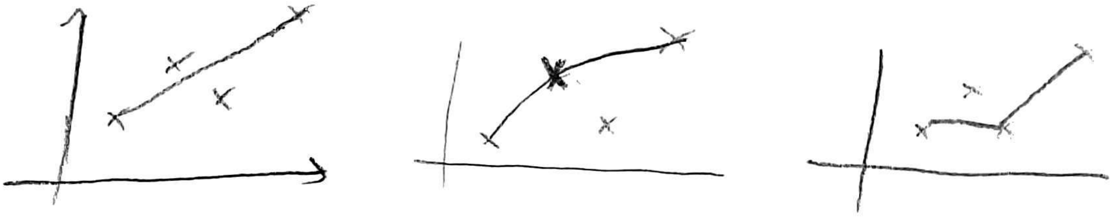


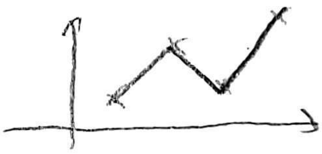
1

n=3 points:

2

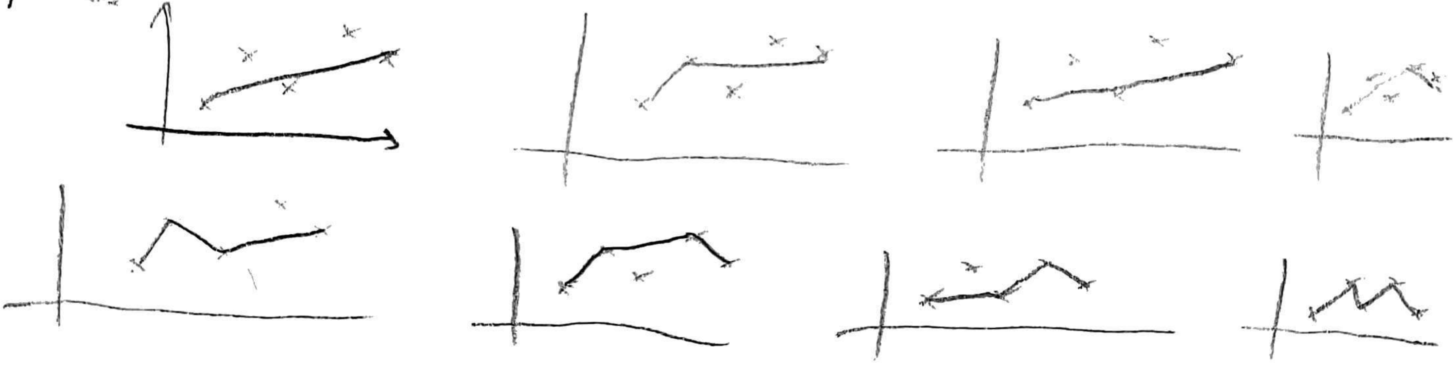
n = 4 points:





4

n= 5 points



8

2^{n-2}

i = i^e point de la liste