TD d'Algorithmique et structures de données
Galerie d'art

Équipe pédagogique Algo SD

L'effervescence est à son comble dans la capitale : le musée du Louvre va ouvrir une nouvelle salle, contenant des œuvres d'art magnifiques. Afin de se prémunir des convoitises des voleurs, le musée a décidé de placer des caméras dotées de la dernière technologie dans la salle, afin de surveiller ses moindres recoins. Chaque caméra sera placée dans un coin de la salle, et peut voir toute partie de la salle qui n'est pas cachée par un mur (voir Figure 1). Néanmoins, ces caméras ont un coût unitaire exorbitant, et le budget du musée est limité en ces temps de crise. Le Louvre fait appel à vos talents d'algorithmicien afin de répondre aux deux questions suivantes :

1. Combien de caméras faut-il pour couvrir toute la salle ?
2. Où placer ces caméras ?

Serez-vous capable d'aider le Louvre à protéger ses œuvres d'art ?

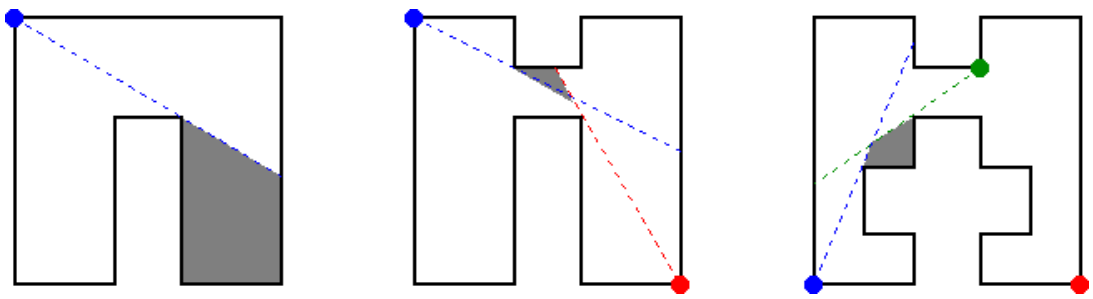

FIGURE 1 – Exemples de galeries surveillées par des caméras, avec un nombre insuffisant de caméras : certaines parties (grisées) sont cachées par des murs. La galerie est représentée en vue du dessus par un polygone, et les caméras (représentées par des points) sont placées sur des sommets du polygone, c'est-à-dire dans des coins de la salle.

1 Formalisation du problème

Le problème de la galerie d'art se formalise de la façon suivante. Un *polygone* est une figure géométrique plane formée d'une suite cyclique de segments consécutifs et délimitant une portion du plan. Un point d'intersection de deux segments consécutifs est appelé *sommet* du polygone. Exemples : triangle, carré, losange, parallélogramme, pentagone, hexagone, ...

Soit P un polygone simple (c'est-à-dire sans auto-intersection et sans trou). Un ensemble S de sommets du polygone *surveille* P si pour tout point p à l'intérieur de P, il existe s dans S tel que le segment $[sp]$ est entièrement inclus dans P.

La figure 1 montre trois contre-exemples, avec respectivement 1, 2 et 3 sommets dans S. Dans chaque cas, il existe des points p (les points de la zone grisée) pour lesquels il n'existe pas de sommet s de S tel que $[sp]$ est entièrement inclus dans le polygone.

1. Pour chacun des trois polygones de la figure 2, donner le nombre nécessaire et suffisant de caméras pour sa surveillance. Justifier (vous pouvez par exemple dessiner une solution).

   Théorème (Václav Chvátal, 1973). *Soit n le nombre de sommets d'un polygone P simple du plan. $\lfloor \frac{n}{3} \rfloor$ sommets sont suffisants pour surveiller P (rappel : $\lfloor x \rfloor$ désigne la partie entière inférieure de x).*

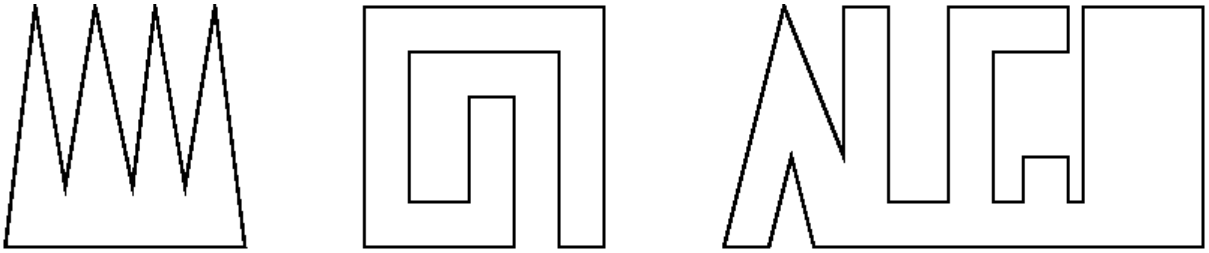

FIGURE 2 – Quel est le nombre minimal de caméras suffisantes pour surveiller chacune de ces trois galeries ?

2. Donner un exemple, avec $n = 6$, pour lequel $\frac{n}{3} = 2$ sommets sont nécessaires. *Indice : vous pouvez vous inspirer d'un des polygones donnés dans les figures précédentes.*

   Une démonstration constructive du théorème de Chvátal, dûe à Steve Fisk (1978) est la suivante (voir Figure 3) :

   1. ajouter des arêtes à l'intérieur de P afin de le découper en triangles ;
   2. attribuer une couleur parmi $\{\texttt{rouge}, \texttt{vert}, \texttt{bleu}\}$ à chaque sommet de P, de manière à ce que pour tout triangle les sommets soient de couleurs différentes ;
   3. déterminer parmi les trois couleurs celle qui est la moins attribuée aux sommets de P, et placer une caméra en chacun de ces sommets.

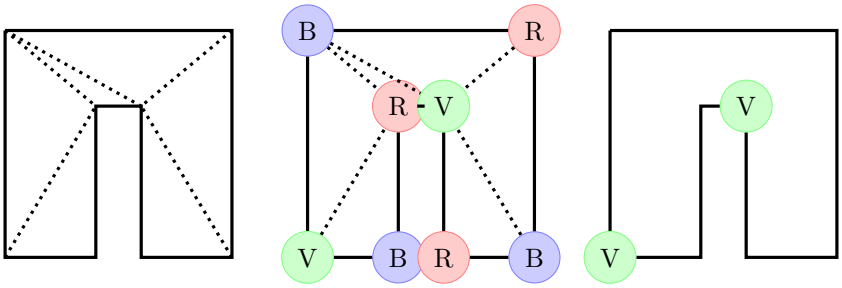

FIGURE 3 – Etapes de la démonstration constructive de Fisk. Ici, deux caméras sont suffisantes : il suffit de les placer sur les sommets verts.

3. Quels triangles surveille chaque caméra ? Expliquer pourquoi cette démonstration justifie la borne $\lfloor \frac{n}{3} \rfloor$ donnée par Chvátal.

2 Placement des caméras

David Avis et Godfried Toussaint ont proposé en 1981 un algorithme "Diviser pour régner" qui permet de trouver où placer les caméras, c'est-à-dire quels sont les sommets de la couleur la moins attribuée. Cet algorithme prend en entrée le polygone P à la fin de l'étape 1. de Fisk, c'est-à-dire **avec les arêtes intérieures le découpant en triangles** (comme sur la Figure 3 à gauche).

L'**étape de récursion** consiste à diviser le polygone P en deux sous-polygones P_1 et P_2. Pour cela, on cherche une arête intérieure $[s_1 s_2]$ reliant deux sommets de P, et on prend pour P_1 le sous-polygone situé d'un côté de l'arête, et pour P_2 le sous-polygone situé de l'autre côté.

L'**étape de fusion** des résultats consiste à modifier les couleurs des sommets de P_2 afin que la couleur de s_1 dans P_2 soit la même que dans P_1, et de même pour s_2. Par exemple, si s_1 est rouge pour P_1 et vert pour P_2 et si s_2 est vert pour P_1 et bleu pour P_2, alors dans P_2 : tous les sommets verts deviennent rouges, tous les sommets bleus deviennent verts, et tous les sommets rouges deviennent bleus.

2.1 Etape de fusion : harmonisation des couleurs

1. Soient $n+1, n_1+1$ et n_2+1 le nombre de sommets de P, P_1 et P_2 respectivement (noter $n+1$ le nombre de sommets de P plutôt que n, et de même pour P_1 et P_2, permet de simplifier les calculs). Soit $C(n)$ la complexité en pire cas de l'algorithme. Donner $C(n)$ en fonction de $C(n_1)$ et $C(n_2)$. On admettra ici que la recherche du segment $[s_1 s_2]$ se fait en temps n, mais vous devez évaluer la complexité en pire cas de l'étape de fusion des résultats.
2. Pour quelles valeurs de n_1 et n_2 $C(n)$ est-il maximum ? Justifier et donner un grand O pour $C(n)$ dans ce cas, en fonction de n, et quand n tend vers l'infini.
3. De la même manière, pour quelles valeurs de n_1 et n_2 $C(n)$ est-il minimum ? Justifier et donner un grand Θ pour $C(n)$ dans ce cas, en fonction de n, et quand n tend vers l'infini.

2.2 Etape de récursion : division du polygone

L'algorithme de division du polygone P d'Avis et Toussaint permet d'obtenir n_1 et n_2 supérieurs ou égaux à $\lfloor \frac{n}{4} \rfloor$. Il divise les sommets de P en 4 sous-ensembles de sommets successifs A, B, C et D, chacun de taille au moins $\lfloor \frac{n}{4} \rfloor$.

4. Justifier qu'il existe au moins une arête intérieure de P qui relie soit A à C, soit B à D. Donner un algorithme simple en temps linéaire pour trouver une telle arête. On admettra que le nombre d'arêtes intérieures est $\Theta(n)$. Rappel : P est un polygone sans trou, et est découpé en triangles grâce aux arêtes intérieures.
5. On suppose n multiple de 4, par souci de simplicité. Déduire de ce qui précède que $C(n) \leq C(\frac{n}{4}) + C(\frac{3n}{4}) + 2n$.
6. Montrer par récurrence sur n que $C(n) \leq dn \log n$, avec d une constante positive bien choisie.
   L'algorithme d'Avis et Toussaint est donc en $O(n \log n)$.

3 Triangulation de polygones

On s'intéresse maintenant au problème de triangulation d'un polygone simple, c'est-à-dire la première étape de la preuve de Fisk.

Le problème est le suivant : on veut décomposer un polygone P en un ensemble de triangles d'intérieurs disjoints dont les arêtes sont des diagonales de P. Une diagonale d'un polygone P est définie comme un segment contenu dans l'intérieur de P et dont les extrémités sont des sommets de P. Une triangulation peut donc être définie par l'ensemble des diagonales qui la composent.

Supposons que l'on dispose d'une fonction permettant de trouver une diagonale dans un polygone P en un temps linéaire $O(n)$.

1. Décrire un algorithme trouvant une triangulation d'un polygone P en utilisant une approche "Diviser pour régner".
2. Ecrire la relation de récurrence qui calcule le coût de cet algorithme. Evaluer le coût en pire cas.

En 1981, Bernard Chazelle prouve le théorème suivant, et ramène la complexité en pire cas du problème de triangulation à $O(n \log n)$:

(polygon cutting, Chazelle 1982) 1. *Soit P un polygone simple à n sommets. Il existe un algorithme de complexité $O(n)$ pour calculer une diagonale de P qui coupe P en deux polygones P_1, P_2 tels que*

$$|P_1| \leq |P_2| \leq \left\lceil \frac{2}{3} n \right\rceil + 1$$

3. Montrer par récurrence que l'utilisation d'un tel algorithme entraine une complexité en pire cas de $O(n \log n)$.

NB : Cet algorithme n'est pas l'état de l'art en matière de triangulation. En 1991, Chazelle montre que l'on peut trianguler un polygone simple en temps linéaire, avec un algorithme réputé très complexe, qui sera simplifié et randomisé par Amato et al. en 2001.