Des Espions au téléphone
Année 2018 – 2019

Elèves : Gaël Bouveron, Théo Pellarin, Julien Ribiollet, élèves de la classe de Terminale S2

Encadrés par : Mme Dauvier, Mme Castillon-Lemettais, Mme Pégeot

Établissements : Lycée Charles Baudelaire (Cran-Gevrier) en collaboration avec le collège Ernest Perrier de la Bathie (Ugine)

Chercheuse : Ariadna Fossas-Tenas, de l'Université de Genève (Suisse)

I/ Le problème

1) L'énoncé du problème

Soit un nombre n d'espions.
Chaque espion, en mission dans un endroit différent, doit récupérer une information secrète.
Une action commune sera alors engagée par tous les agents secrets, une fois qu'ils auront tous récupéré les informations de tous leurs collègues. Pour communiquer leurs informations, les agents ont la possibilité d'appeler par téléphone UN SEUL agent à la fois, et durant cette communication, les deux correspondants se partageront toutes les informations dont ils disposent.
L'objectif est donc de savoir combien d'appels au minimum sont nécessaires pour que tous les espions puissent détenir toutes les informations. Cependant comme le sujet précise qu'il y a n espions nous devons déterminer une formule générale permettant d'exprimer le nombre d'appels minimum en fonction du nombre d'espions.
A noter qu'un espion peut appeler un autre espion plusieurs fois.

2) Conjecture et résultats obtenus

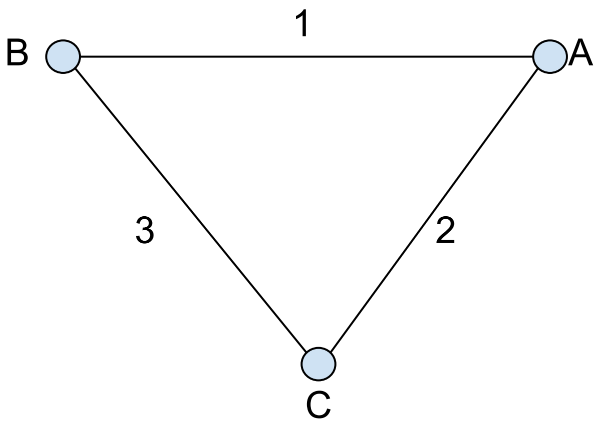
Nous avons d'abord conjecturé puis démontré qu'il suffit de $2n-4$ appels pour que toutes les informations soient transmises, mais nous ne pouvons pas affirmer qu'il s'agit du nombre minimum.
Dans la suite de cet article, nous allons expliquer ce qui nous a amené à affirmer ces résultats.

II/ Résultats expérimentaux

On appelle ici "Résultats expérimentaux" le nombre d'appels minimum que nous avons réussi à obtenir grâce aux méthodes présentées ci-après.

1) Schématisation du problème

Il est possible de représenter géométriquement le problème : on place dans un plan n points (nommés A, B, C, ...) correspondants aux n espions et on représente chaque appel par des segments reliant les deux espions concernés.
On numérote chaque segment pour indiquer l'ordre des appels (cet ordre étant important car un agent ne détiendra pas les mêmes informations s'il a appelé certains de ces collègues en premier ou non).



Représentation géométrique des appels pour n=3

Afin de
représenter les informations détenues par chaque espion, nous avons créé des tableaux les présentant à chaque étape. Les informations portent les mêmes noms que l'espion, comme suit :

	Informations détenues par l'espion A	Informations détenues par l'espion B	Informations détenues par l'espion C
Etat initial	A	B	C
Appel n°1	**AB**	**AB**	C
Appel n°2	**ABC**	AB	**ABC**
Appel n°3	ABC	**ABC**	**ABC**

Informations détenues par chaque espion ($n = 3$) d'après le modèle ci-dessus

2) Résultats

Nombre n d'espions	2	3	4	5	6	7	8	9
Nombre d'appels trouvés	1	3	4	6	8	10	12	14

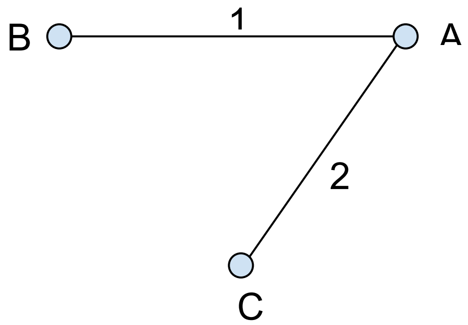
Nombre d’appels trouvés expérimentalement(1)

Nous ne pouvons pas affirmer que ces valeurs sont les nombres minimums d’appels cherchés pour $n \geq 4$ car ces valeurs restent expérimentales et nous n’avons pas pu essayer toutes les configurations (quels appels et dans quel ordre), d’autant plus qu’à mesure que n augmente, le nombre de configurations augmente considérablement(2).

Cependant, nous pouvons dire que les valeurs pour $n = 2$ et $n = 3$ sont les nombres d’appels minimums pour que toutes les informations soient communiquées :

- pour $n = 2$: le seul contact possible est entre les deux espions, et une fois ces contacts effectués, ils détiennent tous deux toutes les informations

- pour $n = 3$: un contact ne suffit pas car un espion n’aurait alors communiqué avec personne et deux contacts ne suffisent pas non plus, comme l’indique le dessin suivant



L’espion B ne connaît pas l’information C

Nous sommes donc sûrs qu’il est nécessaire de passer au minimum 1 appel si nous avons 2 espions et 3 appels si nous avons 3 espions(3).

III/ Démarche de recherche du nombre minimum

Dans cette partie, nous cherchons à vérifier mathématiquement si les valeurs trouvées expérimentalement correspondent aux nombres d’appels minimums.

1) Nombre d’appels nécessaires

Nous allons dans cette partie déterminer un nombre minimum d’appels nécessaires pour n’importe quel nombre d’espions donné, mais ce nombre d’appels peut ne pas être suffisant pour permettre aux espions d’avoir toutes les informations.

Nous savons que les espions peuvent obtenir des informations uniquement en contactant un autre espion. Par conséquent, il est nécessaire que tous les espions communiquent au moins une fois.

Cela implique qu’il faut $n - 1$ contacts, puisqu’ainsi, chacun a appelé au moins une personne.

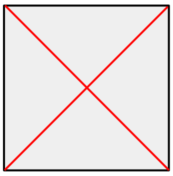
2) Graphes complets

De manière évidente, si tous les espions contactent tous les autres, alors chacun détiendra toutes les informations. Cela revient à ce que chaque espion ne communique que sa propre information, et pas celle des autres.
Géométriquement, cela revient à dénombrer le nombre de segments différents que l'on peut tracer dans un polygone à n sommets distincts.

Conjecture : Soit (U_n) la suite qui dénombre le nombre de segments différents dans un polygone à n sommets, avec $n \geq 4$.
On conjecture que : $U_n = \frac{n(n-1)}{2}$

Démonstration : On s'intéresse ici au nombre de diagonales V_n avec $n \geq 4$.
On cherche à prouver que le nombre de diagonales vaut : $V_n = \frac{n(n-3)}{2}$. Pour cela, nous allons utiliser une démonstration par récurrence(4).

Initialisation ; pour $n = 4$: $V_4 = \frac{4\times1}{2} = 2$
On peut tracer 2 diagonales dans un polygone à 4 sommets :



Hérédité : Soit un entier $n \geq 4$.
On admet qu'un polygone à n sommets a exactement $\frac{n(n-3)}{2}$diagonales.
On cherche à montrer qu'un polygone à $n+1$ sommets a exactement $V_{n+1} = \frac{n(n+1)(n+1-3)}{2}$ diagonales.
Un polygone à $n+1$ sommets a un sommet de plus qu'un polygone à n sommets. On peut donc le voir comme un polygone à n côtés accolé à un triangle.
Il a par conséquent $\frac{n(n-3)}{2}$ diagonales auxquelles on ajoute toutes celles qu'on peut tracer à partir de ce sommet supplémentaire. Il y a $(n-2)$ telles diagonales (on relie le sommet supplémentaire à tous les sommets sauf aux deux qui lui sont adjacents et à lui-même, car ce sont alors des côtés du polygone). Il faut ajouter le segment servant de côté commun au polygone à n sommets et au triangle.
Il y a donc au total : $\frac{n(n-3)}{2} + (n-2) + 1$
Or, $\frac{n(n-3)}{2} + (n-2) + 1 = \frac{n(n-3)+2(n-2)+2}{2} = \frac{n^2-n-2}{2} = \frac{(n+1)(n-2)}{2}$
La propriété est donc héréditaire.
Conclusion : La propriété est vraie au rang 4 et est héréditaire, donc, d'après le principe de récurrence, la propriété est vraie pour tout entier $n \geq 4$.

Retour à (U_n) : U_n est égal au nombre de diagonales ajouté au nombre de côtés d'un polygone à n sommets (soit n côtés).

Donc : $U_n = \frac{n(n-3)}{2} + n = \frac{n^2-3n+2n}{2} = \frac{n(n-1)}{2}$

Pour conclure, il est possible que tous les espions obtiennent toutes les informations en $\frac{n(n-1)}{2}$ appels, si chaque espion appelle tous les autres (avec qui il n'a pas encore eu de contact).

3) Conjecture

La suite (U_n) est certes une solution qui fonctionne mais elle ne désigne pas le nombre minimal d'appels pour n espions. Par exemple, pour $n = 4$: $U_4 = 6$ alors que nous avons trouvé une solution en 4 appels.

En observant le tableau présentant les résultats, nous pouvons observer que, pour tout $n \geq 4$, il suffirait d'ajouter deux appels lorsqu'on ajoute 1 espion pour que toutes les informations soient transmises.

On peut donc conjecturer que le nombre minimum d'appels, que l'on va noter A_n par la suite, est égal à : $A_n = 2n - 4$ pour tout $n \geq 4$.(5)

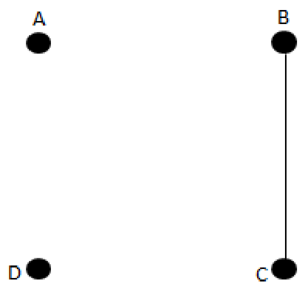
4) Démonstration de la conjecture

Nous utilisons une nouvelle fois une démonstration par récurrence pour montrer que la propriété $A_n = 2n - 4$ est vraie pour tout $n \geq 4$.

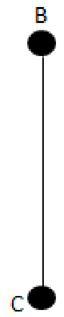
Initialisation : On souhaite montrer que A_4 est le minimum d'appels pour $n = 4$. Pour cela, nous allons montrer qu'il est impossible que les 4 espions possèdent toutes les informations en faisant 1, 2 ou 3 appels, alors que 4 appels sont suffisants.

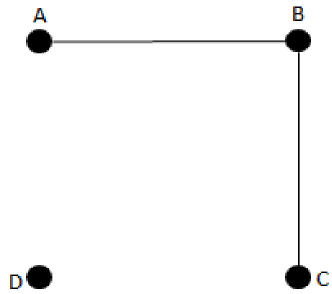
Afin de simplifier la démonstration, nous montrerons simplement à l'aide de schémas que nos affirmations sont correctes. De plus, nous ne considérerons pas la possibilité de faire deux fois le même appel d'affilé.(6)

- Avec 1 appel : On voit que seuls les espions B et C peuvent communiquer entre eux, et donc les espions A et D ne peuvent pas avoir toutes les informations.

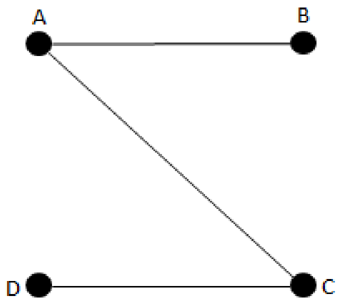


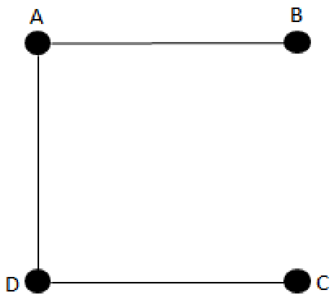
- Avec 2 appels : On voit une fois de plus qu'un espion ne peut pas communiquer avec les autres.

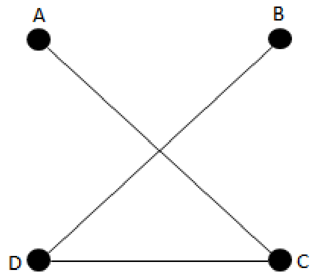


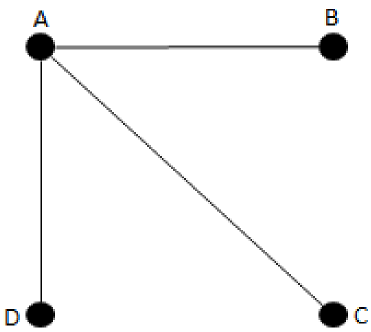


- Avec 3 appels : Nous avons 4 configurations possibles permettant de relier les 4 espions entre eux (il en existe d'autres mais qui laissent au moins un espion de côté) :

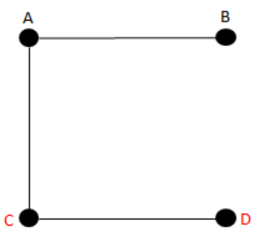


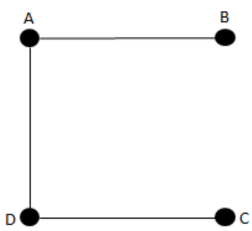


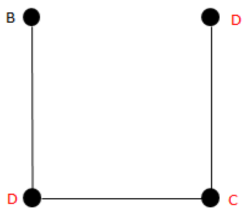




Cependant, on se rend compte que les trois premières figures sont équivalentes : en inversant les espions C et D de la première figure, on retrouve la deuxième, de même en inversant les espions A et B de la troisième figure on obtient la deuxième figure (tournée à 90° vers la gauche) :





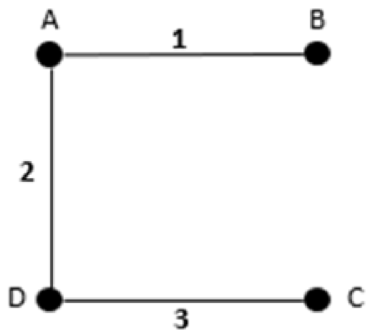


Puisque le nom des espions n'est pas important et que seul la configuration (et l'ordre) des appels est important, il nous suffit d'étudier la figure 2.

Nous allons donc cette fois tester tous les ordres d'appels possibles avec cette configuration, puis avec la deuxième configuration. Pour prouver que le nombre d'appels n'est pas suffisant, nous allons montrer qu'un espion n'a pas toutes les informations à la fin des 3 appels.

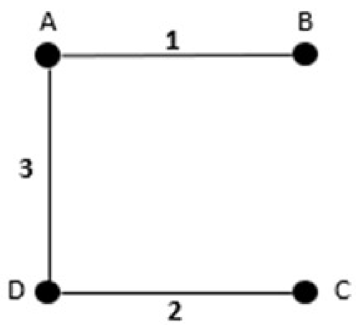
Première configuration :

1. Premier ordre



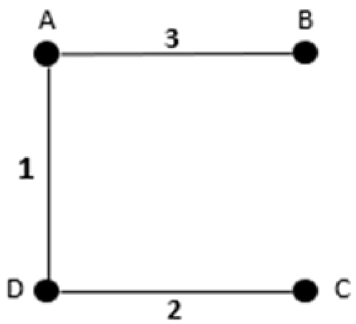
	Espion A	Espion B	Espion C	Espion D
Appel 1	**AB**	**AB**	C	D
Appel 2	**ABD**	AB	C	**ABD**
Appel 3	ABD	AB	**ABCD**	**ABCD**

2. Deuxième ordre



	Espion A	Espion B	Espion C	Espion D
Appel 1	**AB**	**AB**	C	D
Appel 2	AB	AB	**CD**	**CD**
Appel 3	**ABCD**	AB	CD	**ABCD**

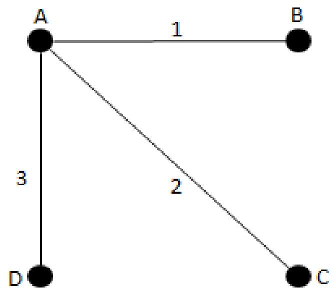
3. Troisième ordre



	Espion A	Espion B	Espion C	Espion D
Appel 1	**AD**	B	C	**AD**
Appel 2	AD	B	**ACD**	**ACD**
Appel 3	**ABD**	**ABD**	ACD	ACD

Nous avons donc essayé 3 ordres différents, mais il en existe d'autre que nous n'avons pas étudiés dans cette partie car ils sont identiques à ceux présentés ci-dessus, par symétrie.(7)

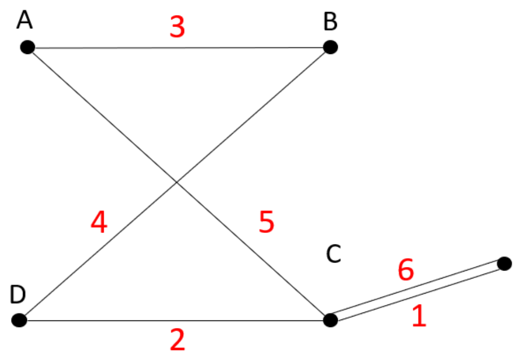
Deuxième configuration :
Cette configuration ci ne permet qu'un seul ordre d'appel possible car il suffit de permuter différents espions pour revenir à celle-ci :



	Espion A	Espion B	Espion C	Espion D
Appel 1	**AB**	**AB**	C	D
Appel 2	**ABC**	AB	**ABC**	D
Appel 3	**ABCD**	AB	ABC	**ABCD**

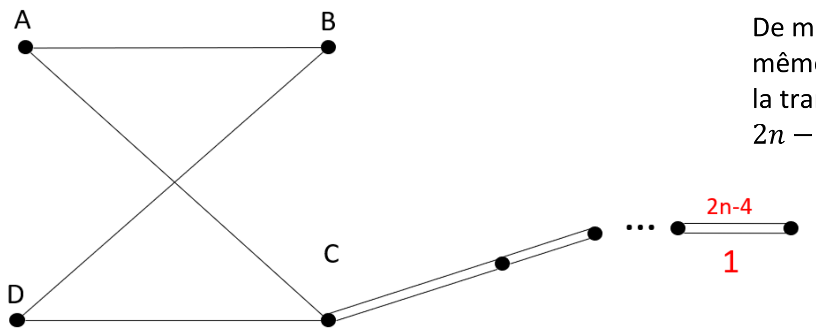
Ainsi, les trois ordres de la première configuration ainsi que l'unique ordre de la deuxième configuration que nous avons essayé ne permettent pas aux espions d'obtenir toutes les informations. 3 appels ne sont donc pas suffisants, et comme nous avons déjà trouvé expérimentalement que 4 appels suffisent, on en déduit que le nombre minimum d'appels nécessaires pour 4 espions est de 4 appels.
Nous avons bien $A_4 = 4$ donc l'initialisation est vérifiée.

Hérédité : Soit un entier $n \geq 4$.
On admet que la propriété est vraie au rang n, c'est à dire $A_n = 2n - 4$. On cherche à montrer que la propriété est vraie au rang $n + 1$, c'est à dire $A_{n+1} = 2(n + 1) - 4 = 2n - 4 + 2 = A_n + 2$.
Lorsqu'on ajoute un espion, il suffit d'ajouter un contact entre ce nouvel espion et la chaîne formée à n espions, puis former la chaîne à n espions, puis refaire le premier contact.
De cette façon, le nouvel espion transmet son information, puis toutes les informations sont transmises comme précédemment (car le modèle avec n espions permet la transmission de toutes les information). Puis le dernier contact permet au nouvel espion de récupérer toutes les informations qu'il n'avait pas encore.



Exemple pour $n = 5$ espions.

On reconnaît la configuration pour n=4 admise dans l'initialisation (à gauche).



De même, en ajoutant des espions et en utilisant le même procédé on peut créer une figure permettant la transmission de toutes les informations en $2n - 4$ appels.

<u>Conclusion :</u> La propriété $A_n = 2n - 4$ est vraie au rang $n = 1$ et elle est héréditaire. D'après le principe de récurrence, cette propriété est vraie pour tout $n \geq 4$.

Nous pouvons donc affirmer que $A_n$est un nombre suffisant d'appels.
Nous conjecturons que A_n est également nécessaire. Cependant, nous n'avons pas réussi à le démontrer.(8)

IV/ Algorithme

Si la suite définie par $A_n = 2n - 4$ semble donner une réponse satisfaisante au problème, elle ne permet toutefois pas d'être certain que la valeur obtenue pour n espions correspond au strict minimum d'appels. Le plus juste serait de dire que cette suite permet d'obtenir, pour n espions, un nombre d'appels faible, proche du nombre minimal d'appels.

Néanmoins, il faut reconnaître à cette suite qu'elle résiste bien à tous les tests expérimentaux auxquels nous l'avons soumise : nous n'avons toujours pas trouvé de contre-exemple prouvant qu'elle ne donne pas le minimum d'appels.

Nous avons élaboré un programme en langage Python qui, justement, est chargé de trouver, s'il en existe, un contre-exemple à cette suite. Son objectif est simple : pour un nombre d'espions saisi par l'utilisateur, le programme attribue une information à chaque espion et les fait s'appeler au hasard jusqu'à que tous les espions aient toutes les informations. Il compte le nombre d'appels effectués et l'enregistre dans une variable. A chaque simulation, le nombre d'appels effectués est comparé à cette variable, et s'il est inférieur, la variable est mise à jour avec la plus petite valeur. On estime qu'au bout d'un nombre très important de simulations, défini arbitrairement par l'utilisateur, le programme finira par simuler la situation optimale d'appels et par trouver le nombre minimal d'appels pour n espions.

Jusqu'alors, le programme semble confirmer les résultats obtenus avec la suite (A_n), et ce jusqu'à 20 espions pour 10 milliards de simulations effectuées en 5 jours environ.

Vous trouverez ci-dessous le détail du programme avec les commentaires du codeur pour bien le comprendre.(9)

```
import sys
import random
#On demande à rentrer le nombre d'espions et le nombre de tentatives à effectuer
#Plus il y a de tentatives, plus on est précis
#On s'assure qu'il s'agit d'un entier avec int()
nbSpies = int(input("Nombres d'espions ?"))
tentative = int(input("Nombres de tentatives ?"))


minimumSteps = 1000 #Variable stockant le nombre de coup de fil minimal
compteur=0 #Variable permettant de répéter le code autant de fois que le nombre de tentatives indiquées

while tentative>compteur:
    #Boucle qui permet d'effectuer un nombre de coup de fil pour une même série
    #allant de 0 au nombre de connexions possibles entre chaque espion
    for k in range(0, int(((nbSpies*(nbSpies-1))/2))):

        #Le programme choisit deux espions qui vont s'appeler
        firstSpy = random.randint(0,nbSpies-1)
        secondSpy = random.randint(0,nbSpies-1)

        #Si le programme a choisi par erreur de faire s'auto-appeler un espion
        #Cette boucle while palie le problème
        while firstSpy==secondSpy:
            secondSpy = random.randint(0,nbSpies-1)
```

```python

        #tempList est une liste temporaire qui est formé de la concaténation
        #des infos des deux espions en question ; on enlève les doublons d'infos
        #et on la trie dans l'ordre croissant d'infos (pour l'esthétique)
        tempList = allSpies[firstSpy] + allSpies[secondSpy]
        tempList = list(set(tempList))
        tempList.sort()

        #On met à jour la grande liste initiale avec les infos échangées par les
        #espions concernés
        allSpies[firstSpy]=tempList
        allSpies[secondSpy]=tempList

        #Dans le cas où TOUS les espions sont pourvus de TOUTES les infos, on
        #arrête la boucle for avec "break" et on vérifie si le nombre de coups
        #de fil effectués est inférieur à celui des séries précédentes. Si c'est
        #le cas, on met à jour le nombre de coups de fil minimum
        it = iter(allSpies)
        if all(len(l) == nbSpies for l in it):
            if minimumSteps>=k+1:
                minimumSteps=k+1
                finalSpiesState=allSpies
            break
        #Pour un suivi étapes par étapes de l'évolution du nombre minimal de coups
        #de fil, on l'affiche à chaque nouvelle série "While"
        print(minimumSteps)

#On affiche finalement le nombre de coups de fil minimum (minimumSteps)
#précédé de la liste "finalSpiesState", permettant de controler qu'au final,
#tous les espions ont bien toutes les infos.
print(finalSpiesState)
print("Le plus rapide échange téléphonique est de "+str(minimumSteps)+" coups de fil.")

```

V/ Conclusion et résultats

Nous avons ainsi prouvé que la suite (A_n) permet de calculer un nombre d'appels suffisants pour que tous les espions possèdent toutes les informations, pour n'importe quel nombre d'espions supérieur à 4.

Cependant nous ne pouvons pas affirmer que ces valeurs sont minimales car la suite se base sur une seule configuration (qui est possible uniquement à partir de 4 espions). On ne peut pas être certain que cette configuration est la plus optimisée pour un plus grand nombre d'espions, d'autant plus qu'il nous est très difficile de le vérifier. Malgré tout, le programme décrit précédemment a confirmé toutes les valeurs de la suite, jusqu'à n=20.

Théorème : $A_n = 2n - 4$ avec $n \geq 4$ est suffisant

Conjecture : A_n est nécessaire

Nous pouvons finalement déterminer un encadrement du nombre minimum d'appels nécessaires pour que tous les espions aient toutes les informations.

Pour tout $n \geq 4$:

$$n - 1 \leq \text{nombre minimum d'appels} \leq A_n \leq U_n$$

Notes d'édition

(1) Ce tableau présentant les résultats est sûrement le fruit d'une recherche organisée et structurée. Il est fort probable que ce soit elle qui soit à la source des démonstrations qui ont été proposées. Il peut être intéressant dans une première lecture d'aller voir p 6 à 9 comment les différents cas ont été organisés.

(2) Question subsidiaire : qu'est ce que « considérablement » : doubler à chaque étape, multiplier par 10, ... ?

(3) Effectivement les rôles de A, B et C ne dépendent pas de ce qu'ils représentent : le nombre ne dépend pas de l'ordre dans lequel on prend les espions.

(4) La récurrence est initialisée à $n = 4$ mais on peut remarquer que si $n = 2$, $U_2 = 2x(2-1)/2 = 1$: c'est le nombre trouvé dans le II-2 et si $n = 3$, on a $U_3 = 3x(3-1)/2 = 3$ et là encore c'est le résultat trouvé précédemment. Lecteur, lectrice, saurez-vous adapter la démonstration en raisonnant directement sur le nombre de segments joignant les sommets du polygone ?

(5) La conjecture « $A_n = 2n - 4$ » rejoint la remarque faite à la p 2. Les études de cas p 6 à 9 permettent de bien comprendre comment le cas $n = 3$ a été utilisé pour construire le cas général.

(6) Comme pour le cas $n = 3$ (page 3), il faudrait préciser que les rôles sont symétriques et le choix des espions n'a pas d'influence. La similitude des statuts des différents espions semble d'ailleurs au cœur du raisonnement comme c'est dit p 7. Sur le cas des 3 appels, est-ce que la présentation garantit que tous les cas ont bien été étudiés, qu'elle a été exhaustive ?

(7) La remarque est judicieuse. Elle justifie de nombreux développements mathématiques et en particulier l'algébrisation des problèmes. L'utilisation des représentations géométriques est pertinente car elle permet d'organiser et de voir un problème. Elle favorise l'intuition, mais lorsque l'on veut être rigoureux, l'intuition a alors une trop grosse emprise dans les représentations géométriques. La modélisation du problème à l'aide de l'algèbre permet d'éviter certaines évidences. Ces deux modes de pensée sont décrits par Henri Poincaré, dans « La valeur de la science » avec les « analystes » et les « géomètres ». Question subsidiaire (pour ceux qui veulent aller plus loin) : quelle théorie mathématique permet de modéliser cette situation ?

(8) La question de la nécessité est intéressante. Car dans cette démonstration : la condition est minimale sur 4, puisqu'en rajoutant seulement deux liaisons à chaque nouvel espion, on pourrait penser que l'on ne peut en ajouter moins. Comment optimiser encore ces échanges : est-ce en trouvant une nouvelle démonstration ? En montrant que si on enlève une liaison alors il y a un manque ? …

(9) C'est dommage que le programme soit une copie d'écran car il faut le ressaisir pour pouvoir insérer des affichages et bien comprendre ce que contiennent les différentes variables.