JULIEN Anthony
L2 Informatique

18/05/2011

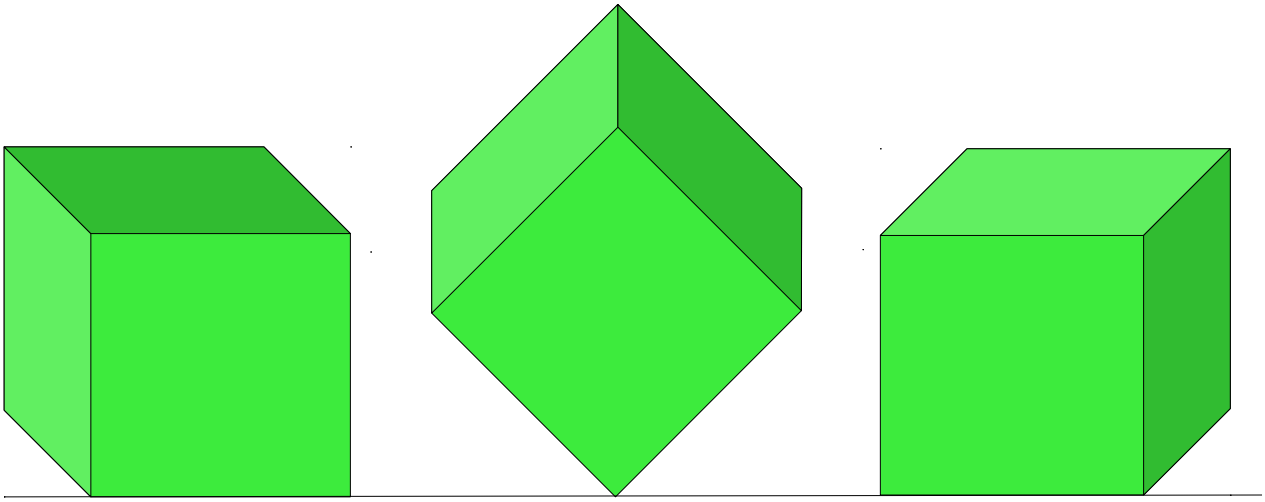
Rapport Individuel Maths en Jeans 1 - 2010 – 2011

Enseignants responsables :

Laurent BEDDOU

Christian MAUDUIT

LE CULBUTO



SOMMAIRE :

1 . Descriptif du Sujet et questionnement_____p.3

2 . Raisonnement et résultats_____p.4

3. Bilan du semestre_____p.11

4. Résolution du défi_____p.12

Descriptif du sujet et questionnent :

Durant la présentation des sujets par notre professeur j'ai de suite était attiré par le thème du culbuto. Non pas que les autres étaient inintéressants, mais celui-ci était le plus amusant et le plus original car je ne savais pas du tout comment on pourrait traiter ce sujet de recherche.

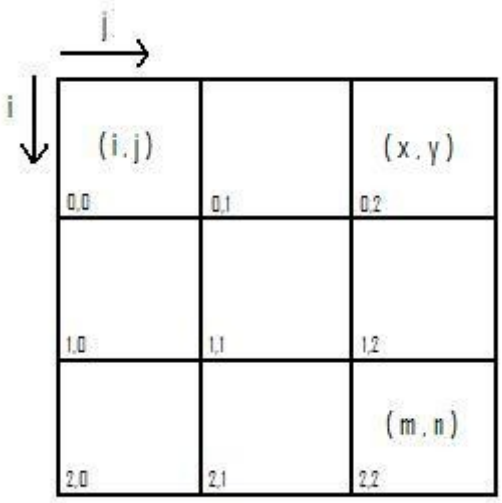
Donc, dans un premier temps j'ai voulu savoir de quoi on allait vraiment parler, et on s'est donc orienter vers l'étude de culbutes de polyèdres. Notre groupe a choisit de travailler plus particulièrement sur les culbutes du cube.

Nous avons choisit de nous représenter les déplacements sous formes de culbutes grâce à un damier. A partir de la, nous nous sommes posés des questions que nous avons traité ensemble ou que j'ai traité individuellement, dont:

- Comment aller d'une case du damier à une autre en autorisant que les culbutes ?
- Combien y a-t-il de chemin permettant cela ?
- Quel est le plus court chemin ?
- Peut-on simuler un glissement sur un damier ?
- Peut-on se servir du culbuto pour vérifier certaines caractéristiques sur un graphe ?
- Quelles sont les différentes façons d'aller d'un point jusqu'au même avec un nombre de culbute précis ?

Raisonnement et résultat:

Pour répondre à la première question, qui est « Comment aller d'une case du damier à une autre en autorisant que les culbutes ? », j'ai trouvé très évident le fait qu'à partir d'une case on pouvait atteindre n'importe quelle autre si on ne mettait aucune contrainte.



représentation de la numérotation
des cases sur un damier

La formule suivante permet de calculer le nombre de chemin d'une case (i,j) à une case (m,n).

$$\binom{(m-i)+(n-j)}{m-i} = \binom{(m-i)+(n-j)}{n-j} = \frac{((m-i)+(n-j))\ !}{\binom{(n-j)!}{(m-i)!} \times \left(((m-i)+(n-j))! - \binom{(n-j)!}{(m-i)!} \right)}$$

Dans un premier temps il faut trouver cette égalité :

$$\binom{n}{p} = \binom{n+1}{p} + \binom{n-1}{p-1}$$

et on a :

$$\binom{n+1}{p} + \binom{n-1}{p-1} = \frac{(n-p) \times (n-1)!}{(n-p)p! \times (n-1-p)!} + \frac{p \times (n-1)!}{p \times (p-1)! \times \left((n-1) - (p-1) \right)!}$$

$$= \frac{(n-p) \times (n-1)!}{p! \times (n-p)!} + \frac{p \times (n-1)!}{p! \times (n-p)!}$$

$$= \frac{n \times (n-1)!}{p! \times (n-p)!} \quad \boxed{= \frac{n!}{p! \times (n-p)!}}$$

En ce qui concerne le calcul du plus court chemin entre une case (i,j) et une case (m,n), on peut facilement en déduire que c'est la différence des coordonnées, c'est-à-dire :

$$\left((m-i) + (n-j) \right)$$

Si on rajoute comme contrainte de passer par une case intermédiaire, il suffit de faire le calcul en deux étapes.

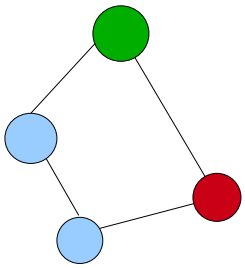
$$\left((x-i)+(y-j) + (m-x)+(n-y) \right)$$

De part l'observation, on détecte clairement que toutes ses formules deviennent fausses lorsque l'on souhaite rajouter la contrainte de terminer sur la même face que celle de départ. En effet, il faut au moins quatre déplacements sur une même colonne ou ligne pour retomber sur la même face.

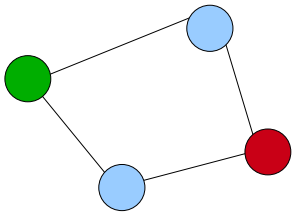
Pour pouvoir simuler un glissement sur un damier en gardant la même orientation, je suis obligé de rajouté de nouveaux éléments. On se repère : pour aller vers le haut on utilise le symbole H, vers le bas B, vers la gauche G, vers la droite D.

Avec une méthode de tâtonnement, on arrive à ces conclusions, pour aller en haut à gauche il faut effectuer cette combinaison B,G,N,D,H,G, en haut à droite B,D, H,G,H,D, en bas à gauche G,B,D,B,G,H, en bas à droite D,B,G,B,D,H.

Afin de faire un lien entre ce cours, et un notre cours de fondement de l'informatique, je me suis intéresser au fait d'utiliser le culbuto pour savoir si un graphe connexe était satisfiable 3-coloration, c'est-à-dire que tous les sommets du graphe doivent avoir une couleur, et il ne faut pas que deux sommets consécutifs soit de la même couleur.
Pour cela il faut mettre nos trois couleurs sur notre cube, et il faut que chaque face opposée soit de même couleur.
A partir de là, on se sert de notre cube comme tampon qui laisserait sa couleur sur le sommet sur lequel il est posé, de cette façon il est facile de créer soit même un graphe 3-colorations ou de vérifier si un graphe existant est satisfiable.



Notre culbuto vérifie que la culbute pour passer d'une face bleue à une autre en une culbute est impossible, ce graphe n'est donc pas satisfiable.



Ici, notre culbuto permet de vérifier que notre graphe est satisfiable 3-coloration.

Enfin, j'ai réfléchi au problème de calculer le nombre de chemins différents permettant de partir d'une case en d'en revenir, avec des contraintes différentes.

Dans un premier temps j'ai cherché à me mettre dans la situation avec un damier 2x2 et en autorisant les oscillations, c'est à dire de repasser par la case ciblée même avant d'avoir fini le chemin en un nombre de coup précis.

Afin de me faciliter la tache j'ai décidé d'écrire un programme qui permet d'envisager tous les chemins possibles et de les compter.

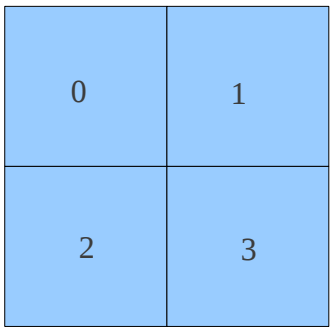
Voici le programme suivi de l'explication :

*2x2.c

```
5 {
6     int T1[1000000],T2[1000000],n,i,j,k,c=0;
7     printf("Donnez le nombre de coups : \n");
8     scanf("%d",&n);
9     for (i=0;i<1000000;i++)
10        T1[i]=T2[i]=-1;
11    T1[0]=0;
12    for (i=0;i<n;i++)
13    {
14        k=0;j=0;
15        while ( T1[k] != -1 )
16        {
17            switch (T1[k] ){
18                case 0 : T2[j]=1;T2[j+1]=2;j+=2;break;
19                case 1 : T2[j]=0;T2[j+1]=3;j+=2;break;
20                case 2 : T2[j]=0;T2[j+1]=3;j+=2;break;
21                case 3 : T2[j]=1;T2[j+1]=2;j+=2;break;
22                }
23            k++;
24        }
25        for (k=0;k<1000000;k++){
26            T1[k]=-1;
27            T1[k]=T2[k];
28            T2[k]=-1;}
29        k=0;
30    }
31    j=0;
32    while ( T1[j] != -1 ){
33        if ( T1[j] == 0 )
34            c++;
35        j++;
36    }
37    printf("il y a %d   solutions\n",c);
38    return 0;
```

Le cœur du programme se trouve à partir de la ligne 17, qui permet de renvoyer les différentes cases où l'on peut faire une culbute a partir de la case où on est en ce moment.

C'est-à-dire :



De la case 0, on peut aller en 1 ou 2.

De la case 1, on peut aller en 0 ou 3 et ainsi de suite.

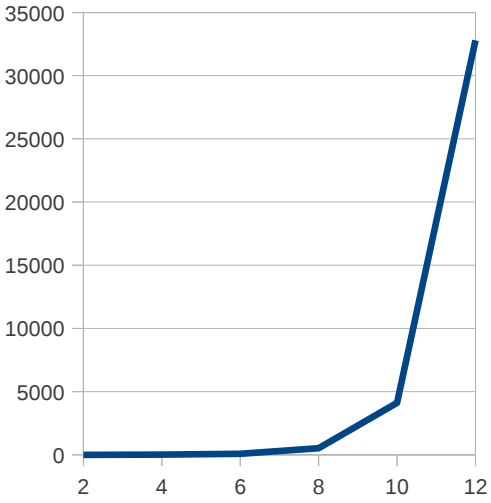
Il suffit juste de faire une boucle selon le nombre de culbute demandé et de versifier si c'est possible ou pas, et de compter tous les chemins.

Cela donne par exemple :

A partir de là j'ai modifié mon programme pour qu'il fonctionne pour de plus grandes tailles de damiers, puis récolté les résultats pour les observer.

Exemple pour un damier 3x3 :

nb culbutes	3x3
2	2
4	10
6	68
8	520
10	4112
12	32800
14	**> Int_Max**



J'ai pu constater deux choses, la première c'est qu'on voit qu'on atteint très vite la valeur de calcul maximale appelée int max, donc la méthode par programmation est moins puissante que de trouver une formule mathématique, c'est pourquoi Julian en a cherché une.
Deuxième chose, on s'aperçoit que les résultats sont toujours pairs, ce qui est logique et démontrable :
Soit un chemin de taille k qui permet de rejoindre une case A à une case B avec un nombre de culbute C.
Si on veut rejoindre A, il faut refaire le même nombre de culbute, ainsi qu'un chemin de taille k.
On se retrouve avec un chemin de taille 2k pour revenir au même point.
2k est multiple de 2, donc pair.

Ensuite, j'ai cherché à observer ce qu'il se passe sur un damier infini, si on s'interdit d'osciller, et qu'on s'oblige que le trajet aller soit le même que le retour.
J'ai essayé de trouver une formule par récurrence.
Pour un nombre de culbute de 2, on trouve 4, car on peut aller au N,S,E,O.

Donc on a:

$$U_2 = 4$$
$$U_4 = 4*(U_2 - 1)=12$$
$$U_6 = 4*(U_4 - 1)=44$$
$$\vdots$$
$$U_{2n} = 4*(U_{2n-2} - 4)$$
$$U_{2n+4} = 4 * U_{2n-2}$$
$$U_2+U_4+U_6+...+U_{2n} = 4(U_2+U_4+U_6+...+U_{2n})-4*(n-2)$$
$$U_2+U_4+U_6+...+U_{2n} = \frac{4*(n-2)}{3}$$

Bilan du semestre :

Le moins que je puisse dire c'est que cette matière est très originale. Pour ce semestre, j'avais le choix entre trois matières de maths différentes, et la seule dont je me poser des questions était celle-ci.
En lisant le descriptif sur internet je m'attendais donc à une unité du style TPE comme au lycée. Mais en fait c'est complétement différent, le fait qu'il y ai deux cours en un, du fait qu'une première partie était consacrée à une explication de différents thèmes mathématiques intéressants avec souvent un support vidéo qui permet d'être plus attractif, et une seconde partie qui consiste à se mettre dans la peau d'un chercheur en travaillant en groupe sur le même thème de recherche.

Sur la partie cours j'ai tout à fait apprécié le fait que chaque semaine c'était un nouveau cours complétement différent de la semaine précédente, ce qui est rare par rapport aux autres matières qui deviennent très vite redondantes.
De plus toutes les activités extérieures m'ont beaucoup intéressé car elles étaient très accessibles, faites par des jeunes dynamiques qui étaient très investis.
L'activité qui m'a le plus surpris car je m'attendais à une conférence plutôt barbante est celle sur les mathématiques Babyloniennes. Cette conférence était très instructive, et on sentait que la chercheuse qui nous parlait était complètement incollable sur son sujet.

Pour ce qui est de la partie travail de groupe et recherche, j'ai été dans un premier temps enthousiaste car c'était une première pour moi, et j'étais attiré par le fait que l'on soit totalement libre. Mais malgré un très bon groupe de travail, je me suis aperçu que cette liberté devenait très vite un inconvénient, on avait tellement de choix possibles dans notre recherche que on ne savait pas par quoi commencé, ni si telle direction nous envoyait dans une impasse.
Et se mettre d'accord sur les questions dont on se posait était une rude tâche, mais toute fois très intéressante.

Enfin si je devais refaire cette unité je la referais car c'est un peu un « ovni » par rapport à nos matières habituelles, le seul point négatif que je peux dire, mais je sais que ça serait difficile de le corriger, c'est le fait que l'on est pas assez suivit pendant les séances de TD, car vous êtes seulement deux professeurs pour une dizaine de groupes de recherche et parfois lorsque vous venez nous voir juste à la fin, nous n'avons pas le temps d'être totalement aiguillé ou aidé.

Résolution du défi :

Un homme à la mer !

Ce schéma permet de visualiser la situation :

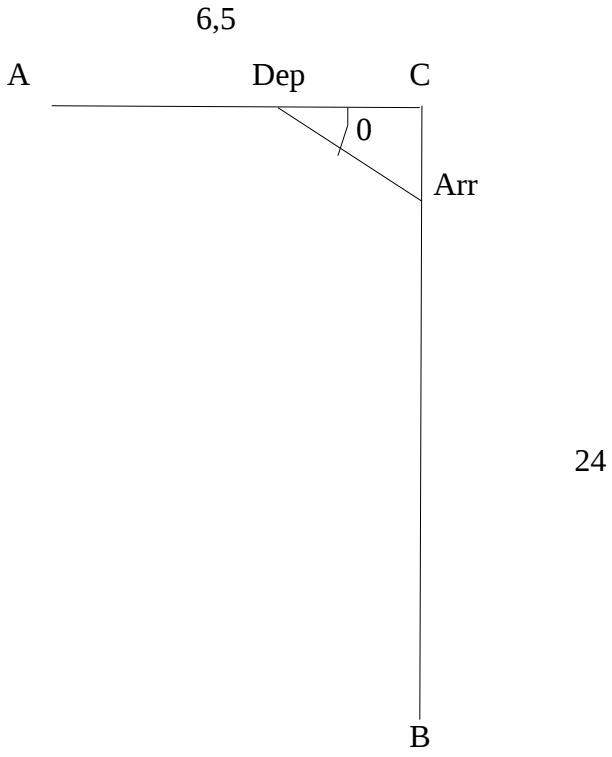
Albatros = A.

Bikini = B.

collision = C.

Le point de départ du nageur = Dep.

Le point d'arrivée du nageur = Arr.



Soit d= Dep*Arr,

et θ= (AC,MN).

Entre le départ et le moment où le nageur monte dans B il s'est passé :

$$\frac{6{,}5-d\cos\theta}{35}+\frac{d}{1{,}5}=\frac{24-d\sin\theta}{35}$$

donc d = $\dfrac{105}{140+6(\sin\theta-\cos\theta)}$

et , $\sin\theta-\cos\theta=\sqrt{2}\sin\left(\theta-\frac{\pi}{4}\right)$

Si : $\theta = \frac{3\pi}{4}$

Alors, $\sin\theta - \cos\theta = \sqrt{2}$

Donc d min = : $\frac{105\sqrt{2}}{140\sqrt{2}+12} \approx \frac{1}{\sqrt{2}}$

Le temps d'attente avant de plonger est alors :

$$\frac{13+d_{min}\sqrt{2}}{70} \approx \frac{1}{5}$$ d'heure.

Ce qui correspond à environ 12 minutes.