Correction TD 7 – Partiel d'Algorithmique 2003

Exercice 1 – le grand saut

1 - La complexité en $O(\log_2(n))$ qui est indiquée nous aiguille vers une dichotomie. En effet, en supposant que l'on a $k \geq \lceil \log_2(n) \rceil$, on obtient le résultat sur les étages de i à j en jetant un étudiant depuis le $n^{\text{ème}}$ étage, où $n = j - i/2$, puis en itérant le procédé sur les étages de i à $n-1$ si la chute à été fatale, et sur les étages de n à j dans le cas contraire. La méthode dichotomique nous garantit alors que l'on obtient le bon résultat (lorsqu'il ne reste plus qu'un seul étage, c'est-à-dire lorsque l'on calcule pour les étages de $i = j$), et ce avec une complexité logarithmique dans le pire des cas.

2 - Puisqu'ici on ne dispose que de $k < \lceil \log_2(n) \rceil$ étudiants, on ne peut pas appliquer directement la méthode dichotomique proposée précédemment. Cependant, on va remédier au problème de manière simple en appliquant une recherche dichotomique avec $k-1$ étudiants, de manière à délimiter un intervalle d'étages dans lequel se trouve l'étage recherché. On se sert alors du dernier étudiant restant pour parcourir l'intervalle de façon linéaire, donc en le jetant de chaque étage en partant du plus bas de l'intervalle, jusqu'au plus haut. Après avoir jeté les $k-1$ premiers étudiants, si l'on n'a pas encore trouvé le bon étage, il reste exactement $n/2^{k-1}$ étages dans l'intervalle de recherche, d'où une complexité dans le pire des cas en $O(k + n/2^{k-1})$ sauts.

3 - Dans le cas particulier où l'on a $k = 2$, on ne veut pas avoir à tester chaque étage de façon linéaire, c'est pourquoi on va reprendre à notre compte les idées précédentes, et notamment celle qui consiste à délimiter un intervalle de recherche. Nous découpons donc l'ensemble des étages en "tranches" de $\sqrt{n}$ étages, avant de jeter le premier étudiant de chacun des étages de début de tranche. Lorsque l'étudiant y laisse sa peau, on se ramène au dernier étage n testé qui ne soit pas fatal, et on n'a plus qu'à parcourir de manière linéaire l'intervalle allant de l'étage $n+1$ à l'étage fatal trouvé précédemment. On a ainsi deux séries d'essais en $O(\sqrt{n})$, et donc une complexité finale dans le pire des cas également en $O(\sqrt{n})$ sauts.

Exercice 2 – plus grand carré de 1

On considère la matrice A de taille $n \times m$ dont les coeficients valent 0 ou 1, et on note $C_{i,j}$ le plus grand carré ayant l'élément $A_{i,j}$ comme coin supérieur gauche, et $t[i,j]$ sa largeur. On peut alors établir la relation de récurrence suivante :

$$\forall i,j < n,m \quad \begin{cases} \text{si } A_{i,j} = 0, & t[i,j] = 0 \\ \text{sinon}, & t[i,j] = \min(t[i,j+1], t[i+1,j], t[i+1,j+1]) + 1 \end{cases}$$

En effet, le premier cas se vérifie de manière triviale, et l'on peut vérifier le second cas, en remarquant déjà que l'on a $t[i+1,j+1] \geq \max(t[i,j+1], t[i+1,j]) - 1$. Ensuite, si l'on pose $l = t[i,j+1]$ et que l'on a $l < t[i+1,j] \leq t[i+1,j+1] - 1$, alors $A_{i,j+l+1} = 0$ et donc $t[i,j] \leq l+1$. D'où, le carré de coin supérieur gauche $A_{i,j}$ et de côté $l+1$ est inclu dans l'ensemble $C_{i,j+1} \cup C_{i+1,j} \cup \{A_{i,j}\}$, il est donc entièrement constitué d'éléments valant 1. On procede de la même manière si $t[i+1,j] < t[i,j+1]$, ce qui permet de vérifier la formule donnée précédemment. Dans le cas où $t[i+1,j] = t[i,j+1] = l$, on a $t[i,j] = l+1$ si $A_{i+l,j+l} = 1$ (et dans ce cas $t[i+1,j+1] \geq l$) mais sinon, on a $t[i,j] = l-1$ (et dans ce cas $t[i+1,j+1] = l-1$). On retrouve bien notre formule. Tout ceci nous fournit l'algorithme suivant :

```
/* initialisation des bords de la matrices */
pour i de 1 à n faire t[i,m] ← A_{i,m}
pour j de 1 à m faire t[n,j] ← A_{n,j}

/* calcul des t[i,j] dans le bon ordre */
pour i de n−1 à 1 faire
        ▶ pour j de m−1 à 1 faire
                ▶ si A_{i,j} = 0 alors faire t[i,j] ← 0
                  sinon faire t[i,j] ← min(t[i,j+1], t[i+1,j], t[i+1,j+1]) + 1
retourner max_{i,j} t[i,j]
```

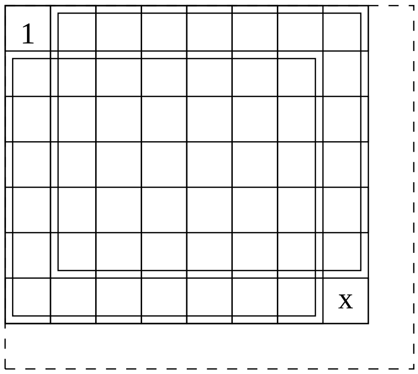

Figure 1: schéma des carrés à prendre en compte pour le calcul de $t[i,j]$

Exercice 4 – cherchez la star

1 - Il ne peut y avoir qu'une seule star dans le groupe, puisque s'il y en a une, elle ne connait personne, et donc tous les autres sont inconnus d'au moins une personne, et ne peuvent donc être eux aussi des stars.

2 - Lorsqu'on effectue le test $''i \rightarrow j\ ?\ ''$, c'est-à-dire lorsque l'on cherche à savoir si la personne i connaît la personne j, on obtient le résultat suivant :

- si oui, alors i n'est pas une star, mais j en est potentiellement une.
- si non, alors j n'est pas une star, mais i en est potentiellement une.

L'algorithme consiste alors a parcourir le tableau des personnes une fois, en gardant en memoire à chaque instant la personne i qui est jusqu'ici reconnue par tous, tandis qu'au $j^{\text{ème}}$ test, toutes les autres personnes, dont les indices sont les $k < j$ (avec $k \neq i$, bien sûr) ne peuvent pas des stars. Cet algorithme s'écrit donc de la manière suivante :

$i \leftarrow 1$ **et** $j \leftarrow 2$
tant que $j \leq n$ **faire**
    ▶ **si** $''i \rightarrow j''$ **alors faire** $j \leftarrow j+1$
    sinon faire $i \leftarrow j$ **et** $j \leftarrow j+1$
$i_{star} \leftarrow$ *vrai* **et** $k \leftarrow 1$
tant que ($k \leq n$ **et** i_{star}) **faire**
    ▶ **si** $''k \rightarrow i''$ **alors faire** $k \leftarrow k+1$
    sinon faire $i_{star} \leftarrow$ *faux*
si i_{star} **alors faire retourner** *“ i est la star ”*
sinon faire retourner *“ il n'y a pas de star ”*

3 - En observant notre algorithme, on constate que l'on peut utiliser comme borne inférieure pour le nombre de questions la valeur $3n-2$, puisque l'on fait ici deux boucles sur $n-1$ personnes et une boucle sur l'ensemble des n personnes. Pour ce qui est de la borne inférieure optimale, il s'agit d'une question difficile, dont on n'explicitera pas la solution ici.