ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE
Section d'Informatique et de Systèmes de Communication

Corrigé de la série 11 13 December 2010

1. La taille de l'input

a) $\log_2(n)$.

b) n^2.

c) $\lceil \log_2(1000) \rceil n = 10n$. (On peut aussi faire légèrment mieux, en représentant les entiers $a_1, \ldots, a_n$ comme un grand entier $\sum_{i=1}^{n} 1000^{i-1} a_i$, ce qui donne $\lceil \log_2(1000) n \rceil$.)

d) $n^2 \cdot \log_2(201+1) = O(n^2)$.
Explication : On a n^2 valeurs dans la matrice d'adjacence, pour chaque valeur on a besoin de $\log_2(201+1)$, bits : 201 pour les valeurs possibles de -100 à 100 et une valeur de plus pour codifier le fait qu'il n'y a pas d'arête entre les deux sommets.

2. Réduction polynomiale

a) Supposons que $G = (V, E)$, et soit u le sommet ajouté pour obtenir $f_1(G)$. Ainsi $f_1(G) = (V_1, E_1)$ où $V_1 = V \cup \{u\}$, et $\forall v \in V_1 : \ (u, v_i), (v_i, u) \in E_1$.

$\Rightarrow$. Supposons que G contient une clique $S \subseteq V$ de taille $\geq k$. Puisque u est connecté à tous les autres sommet,
$$S \cup \{u\} \subseteq V_1$$
forme une clique de taille $|S| + 1 \geq k + 1$.

$\Leftarrow$. Supposons que $f_1(G)$ contient une clique $S \subseteq V_1$ de taille $\geq k+1$. Nous distinguons 2 cas :

• **Cas 1 :** $u \notin S$. Dans ce cas nous avons $S \subseteq V$, et donc cette clique de taille $\geq k+1$ est aussi dans G. G contient donc bien une clique de taille $\geq k$.

• **Cas 2 :** $u \in S$. Si nous retirons u de la clique nous obtenons une clique de taille $|S| - 1$, donc de taille $\geq k$. puisque tous les sommets de V_1 sauf u sont aussi dans V, $S - \{u\} \subseteq V$ forme une clique de taille $\geq k$ dans le graphe G.

b) Nous ajoutons a sommets $u_1, \ldots u_a$, qui sont connectés à tous les autres sommets (et aussi connecté entre eux). Soit $U = \{u_1, \ldots, u_a\}$. Formellement si $G = (V, E)$ alors $f_a(G) = (V_a, E_a)$ où
$$\begin{array}{rcl} V_a & = & V \cup U \\ E_a & = & E \ \cup \ (V_a \times U) \ \cup \ (U \times V_a) \end{array}$$
Nous pouvons montrer de manière identique au point a) que G contient une clique de taille $\geq k$ si et seulement si $f_a(G)$ contient une clique de taille $\geq k + a$.

c) Nous voyons d'abord que
$$n_a = |V_a| = |V| + a = m + a.$$

Nous savons que G contient une clique de taille $\geq k$ si et seulement si $f_a(G)$ contient une clique de taille $\geq k+a$. Nous voulons donc choisir a tel que $n_a/2 = k+a$.

$$\begin{aligned} \tfrac{n_a}{2} = k+a \quad &\iff \quad \tfrac{m+a}{2} = k+a \\ &\iff \quad m+a = 2k+2a \\ &\iff \quad a = m-2k \end{aligned}$$

Puisque $k \leq m/2$, nous voyons que cette valeur est ≥ 0.

d) Il nous reste à couvrir le cas où $k > m/2$. Nous construisons le graphe $h(G)$ en ajoutant à V $2k-m$ sommets sans aucune arête. Formellement, $h(G) = (V', E')$ avec

$$\begin{aligned} V' &= V \cup \{w_1, \ldots, w_{2k-m}\} \\ E' &= E. \end{aligned}$$

Puisque les nouveaux sommets n'on par d'arêtes, il est clair que G contient une clique de taille $\geq k$ si et seulement si $h(G)$ contient une clique de taille $\geq k$.

Ainsi la réduction est la suivante :

$$f(G,k) = \begin{cases} f_{m-2k}(G) & \text{si } k \leq \frac{m}{2} \\ h(G) & \text{sinon} \end{cases}$$

3. L'algorithme de Karp

a) Soit $s \to \cdots \to x$ un chemin de poids minimal et de longueur k. On suppose $k \geq 2$ (l'autre cas est immédiatement réglé), et alors on peut écrire ce chemin $s \to \cdots \to u \to x$, où u est l'avant-dernier sommet sur ce chemin. Le chemin $s \to \cdots \to u$ est de longueur $k-1$ et certainement minimal, parce que sinon on pourrait le remplacer par le chemin minimal et on aura aussi un meilleur chemin de s à x.

Donc par hypothèse d'induction le poids du morceau $s \to \cdots \to u$ est $F_{k-1}(u)$, et le chemin $s \to x$ a certainement poids

$$F_{k-1}(u) + w(u,x).$$

Le reste est évident.

b) Comme le graphe est connexe, certainement $|V| \leq |E|-1$, ce qui montre que F_0 peut bien être calculé en $O(|E|)$ opérations. La suite est par induction sur k.

Supposons qu'on a déjà calculé $F_{k-1}, \ldots, F_0$ en $O(|E|k)$ opérations. Si nous montrons que F_k peut alors être trouvé en $O(|E|)$, nous avons fini. Il s'agit d'appliquer la formule du point a). On procède comme suit :

```
for x ∈ V do
    F_k(x) ← ∞
for (x, y) ∈ E do
    u ← F_{k−1}(x) + w(x, y)
    if u < F_k(y) then
        F_k(y) ← u.
```

Comme $|V| = O(|E|)$, cet algorithme utilise évidemment $O(|E|)$ opérations. Il est correct à cause du point précédent.

c) D'abord on calcule les F_k et ensuite le μ^*. Voir l'Algorithme 1 pour les détails.

Algorithme 1 Karp

```
Choisir $s \in V$ arbitraire.
for $x \in V$ do
    $F_0(x) \leftarrow \infty$
$F_0(s) \leftarrow 0$
for $k = 1, \ldots, n$ do
    for $x \in V$ do
        $F_k(x) \leftarrow \infty$
    for $(x, y) \in E$ do
        $u \leftarrow F_{k-1}(x) + w(x, y)$
        if $u < F_k(y)$ then
            $F_k(y) \leftarrow u$.
$\mu^* \leftarrow \infty$
for $x \in V$ do
    $\mu_{\text{tmp}} \leftarrow -\infty$
    for $k = 0, \cdots, n-1$ do
        $u \leftarrow \frac{F_n(x) - F_k(x)}{n-k}$
        if $u > \mu_{\text{tmp}}$ then
            $\mu_{\text{tmp}} \leftarrow u$
    if $\mu_{\text{tmp}} < \mu^*$ then
        $\mu^* \leftarrow \mu_{\text{tmp}}$
```