Transmission d'information dans les arbres binomiaux

Durée : 2 heures

Définitions

Un arbre $\mathcal{T} = (r, \mathcal{L})$ est défini par la donnée d'un entier r, appelé *nœud*, et d'une liste d'arbres $\mathcal{L}$, éventuellement vide. Si la liste $\mathcal{L}$ est vide, on dit que r est un nœud *externe*. Dans le cas contraire, la liste $\mathcal{L}$ comprend ℓ éléments $\mathcal{T}_i = (r_i, \mathcal{L}_i)$, $0 \leqslant i \leqslant \ell$, on dit que r est un nœud *interne*, que les nœuds r_i sont les *fils* de r, et que r est le *père* des r_i. Enfin, le nœud r sera appelé la *racine* de l'arbre $\mathcal{T}$.

On conviendra que dans un arbre, tous les nœuds sont des entiers *distincts*.

Dans un arbre, les *voisins* d'un nœud sont son père et ses fils (s'ils existent).

Entre deux nœuds s et t de $\mathcal{T}$ il existe un unique chemin, composé de nœuds $x_0, x_1, \ldots, x_k$ deux à deux distincts, tels que $x_0 = s$, $x_k = t$, et x_i et x_{i+1} sont voisins pour $0 \leqslant i \leqslant k-1$. On dit que k est la *longueur* de ce chemin.

La *profondeur* d'un nœud est la longueur du chemin qui le relie à la racine ; en particulier, la racine a pour profondeur 0. La profondeur d'un arbre est le maximum de la profondeur de ses nœuds.

Les arbres seront représentés en Caml de la manière suivante :

```
type arbre = Noeud of int * arbre list ;;
```

Question 1. Écrire une fonction **profondeur** qui prend en argument un arbre $\mathcal{T}$ et renvoie sa profondeur.

```
profondeur : arbre -> int
```

Partie I. Arbres binomiaux

Soit k un entier positif ou nul. Un *arbre binomial* d'ordre k est défini comme suit :

– un arbre binomial d'ordre 0 est réduit à sa racine ;

– si $k > 0$, un arbre binomial d'ordre k est de la forme $\big(r_k, (\mathcal{T}_{k-1}, \ldots, \mathcal{T}_1, \mathcal{T}_0)\big)$, où chaque $\mathcal{T}_i$ est un arbre binomial d'ordre i.

Dans la suite, $\mathcal{B}_k$ désignera un arbre binomial d'ordre k.

Question 2. Dessiner $\mathcal{B}_4$ avec une numérotation des nœuds de votre choix (on rappelle que dans un arbre, les nœuds sont des entiers deux à deux distincts).

Question 3. Quelle est le nombre de nœuds de $\mathcal{B}_k$? Combien sont externes ?

Question 4. Pour $k > 0$, montrer qu'on peut aussi définir récursivement $\mathcal{B}_k$ à l'aide de deux copies de $\mathcal{B}_{k-1}$.

Question 5. Montrer que si $k \geqslant 1$, un arbre binomial d'ordre k à qui on a supprimé les nœuds terminaux se transforme en un arbre binomial d'ordre $k-1$.

Question 6. Écrire une fonction **copie** qui prend en arguments un entier n et un arbre $\mathcal{T}$ et renvoie une copie de l'arbre $\mathcal{T}$ dans laquelle chaque nœud de numéro i est remplacé par un nœud de numéro $i+n$.

```
copie : int -> arbre -> arbre
```

Question 7. Écrire une fonction **bin** qui prend en argument un entier $k \geqslant 0$ et qui renvoie l'arbre $\mathcal{B}_k$, avec une numérotation des nœuds de votre choix. On rappelle que dans un arbre, les nœuds sont des entiers deux à deux distincts.

```
bin : int -> arbre
```

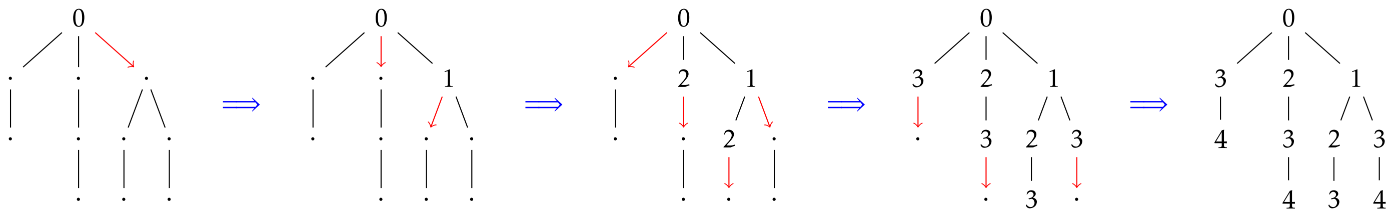
Question 8. Quelle est la profondeur de $\mathcal{B}_k$? Quelle est la longueur maximale d'un chemin entre deux nœuds ?

Question 9. Combien de nœuds ont une profondeur donnée ℓ ?

Partie II. Diffusion dans les arbres

On étudie dans cette partie le problème de la diffusion dans les arbres. La racine r de l'arbre $\mathcal{T} = (r, \mathcal{L})$ possède un message qu'elle doit transmettre à tous les autres nœuds. La diffusion procède par étapes, toutes de temps unitaire. À une étape donnée, chacun des nœuds déjà en possession du message le transmet à un et un seul de ses fils (sauf si tous ses fils l'ont déjà reçu). Une diffusion est donc caractérisée par un ensemble de fonctions : à chaque nœud interne v ayant n_v fils on associe une fonction injective f_v qui numérote ses fils de 1 à n_v, dans l'ordre dans lequel v leur transmet le message.

Voici un exemple de diffusion dans un arbre, dans lequel chaque nœud porte la date à laquelle il a reçu le message :



La *durée* d'une diffusion est le nombre d'étapes nécessaire pour que tout nœud reçoive le message (elle est de 4 dans l'exemple ci-dessus). Une diffusion est *optimale* si sa durée est minimale parmi les durées de toutes les diffusions.

Diffusion dans un arbre binomial

Considérons l'arbre binomial $\mathcal{B}_k$ défini dans la partie précédente. Tout nœud interne v a pour fils les racines $r_1, \ldots, r_{n_v}$ d'arbres binomiaux $\mathcal{B}_0, \ldots, \mathcal{B}_{n_v-1}$. La numérotation naturelle s'obtient en posant $f_v(r_i) = i$, tandis que la numérotation inversée s'obtient en posant $f_v(r_i) = n_v + 1 - i$.

Question 10. Quelle est la durée de diffusion qui choisit la numérotation naturelle pour chaque nœud ?

Question 11. Quelle est la durée de diffusion qui choisit la numérotation renversée pour chaque nœud ?

Question 12. Quelle est la durée d'une diffusion optimale dans $\mathcal{B}_k$? Justifier votre réponse.

Diffusion dans un arbre quelconque

Question 13. Proposer un algorithme pour calculer la durée d'une diffusion optimale dans un arbre $\mathcal{T}$ quelconque (on ne demande pas d'écrire la fonction Caml correspondante).

Question 14. Donner un majorant de la durée d'une diffusion optimale dans un arbre quelconque à n nœuds, et exhiber pour tout n un arbre pour lequel cette borne est atteinte.

Question 15. Donner un minorant de la durée d'une diffusion optimale dans un arbre arbitraire à n nœuds, et exhiber pour tout n un arbre pour lequel cette borne est atteinte.