

Algorithmique — IN102

TD 4

6 janvier 2006

Exercice 1 Voir polycopié.

Exercice 2 Voir polycopié.

Exercice 3

```
// Création d'un tas vide de taille m
tas nouveauTas(int m) {
    tas t = new strTas;
    int *a = new int[m];
    for (int i = 0; i < m; i++)
        a[i] = 0;
    t->nb = 0; t->max = m; t->tab = a;
    return t;
}

// Impression préfixée d'un tas
void imprimeTas(tas t) {
    for (int i = 0; i < t->nb; i++)
        cout << t->tab[i] << " ";
    cout << endl;
}

// Maximum d'un tas
int maxTas(tas t) {
    if (t->nb == 0) {
        cout << "Erreur: tas vide" << endl;
        exit(1);
    } else
        return t->tab[0];
}

// Échange deux éléments passés par référence
void echange(int &a, int &b) {
    int tmp = a; a = b; b = tmp;
}

// Insertion de la clé k dans le tas t
void insereTas(int k, tas t) {
    if (t->nb == t->max) {
        cout << "Erreur: tas plein" << endl;
        exit(1);
    } else {
        int i = t->nb;
        t->tab[i] = k;
        while ( ( i > 0 ) && ( t->tab[i] > t->tab[(i-1)/2] ) ) {
```

```

    echange(t→tab[i], t→tab[(i-1)/2]);
    i = (i-1)/2;
}
t→nb++;
}
}

```

Exercice 4

// Suppression du maximum (renvoyé en résultat)

```

int supprimeTas(tas t) {
    int mx = maxTas(t);
    // On sait maintenant que t n'est pas vide
    t→nb--;
    t→tab[0] = t→tab[t→nb];

    int i = 0;
    int continuer = false;
    do {
        if (2*i + 2 < t→nb) // Noeud à 2 fils
            if ( (t→tab[2*i+1] > t→tab[2*i+2]) && (t→tab[i] < t→tab[2*i+1]) ) {
                // le FG est le max, et la racine est plus petite
                echange(t→tab[i], t→tab[2*i+1]);
                i = 2*i+1;
                continuer = true;
            } else
                if ( (t→tab[2*i+1] ≤ t→tab[2*i+2]) && (t→tab[i] < t→tab[2*i+2]) ) {
                    // le FD est le max, et la racine est plus petite
                    echange(t→tab[i], t→tab[2*i+2]);
                    i = 2*i+2;
                    continuer = true;
                } else // Pas d'échange à faire car la racine est plus grande
                    continuer = false;
        else
            if (2*i + 1 < t→nb) // Noeud à un seul fils
                if (t→tab[i] < t→tab[2*i+1]) {
                    echange(t→tab[i], t→tab[2*i+1]);
                    continuer = false;
                } else // La racine est à sa place
                    continuer = false;
            else // On est sur une feuille
                continuer = false;
    } while (continuer);
    return mx;
}

```

Exercice 5 *L'insertion des n éléments : $n \times \Theta(\log(n))$, puis répéter n fois le calcul du maximum, en $\Theta(1)$, et sa suppression, en $\Theta(\log(n))$. Soit au total une complexité de $\Theta(n \times \log(n))$, en moyenne ainsi que dans le pire des cas.*

Exercice 6

```
// Tri des n premiers éléments du tableau a
void triTas(int *a, int n) {
    tas t = nouveauTas(n);
    for (int i = 0; i < n; i++)
        insereTas(a[i], t);
    for (int i = n-1; i >= 0; i--)
        a[i] = supprimeTas(t);
    delete [] t->tab;
    delete t;
}
```