

## TME 4 - Phase

### Exercice(s)

#### Exercice 1 – L’algorithme PHASE

On veut réaliser un calcul de minimum sur un réseau orienté de topologie quelconque. On va pour cela utiliser l’algorithme PHASE. On utilise toujours le même principe : le processus 0 envoie aux processus 1 à n des informations sur la topologie. Comme le réseau est orienté, les informations fournies distinguent maintenant pour chaque noeud les voisins entrants et les voisins sortants. La fonction simulateur correspondant à la topologie de la figure 1 est donnée par :

```
#define TAGINIT      0
#define NB_SITE 6

#define DIAMETRE      /* !!!!! valeur a initialiser !!!!! */
#define DEG_IN_MAX 2   /* le max des degres entrants des noeuds */
#define DEG_OUT_MAX 2  /* le max des degres sortants des noeuds */

void simulateur(void) {
    int i;

    /* nb_voisins_in[i] est le nombre de voisins entrants du site i */
    /* nb_voisins_out[i] est le nombre de voisins sortants du site i */
    int nb_voisins_in[NB_SITE+1] = {-1, 2, 1, 1, 2, 1, 1};
    int nb_voisins_out[NB_SITE+1] = {-1, 2, 1, 1, 1, 2, 1};

    int min_local[NB_SITE+1] = {-1, 4, 7, 1, 6, 2, 9};

    /* liste des voisins entrants */
    int voisins_in[NB_SITE+1][2] = {{-1, -1},
                                     {4, 5}, {1, -1}, {1, -1},
                                     {3, 5}, {6, -1}, {2, -1}};

    /* liste des voisins sortants */
    int voisins_out[NB_SITE+1][2] = {{-1, -1},
                                     {2, 3}, {6, -1}, {4, -1},
                                     {1, -1}, {1, 4}, {5, -1}};

    for(i=1; i<=NB_SITE; i++){
        MPI_Send(&nb_voisins_in[i], 1, MPI_INT, i, TAGINIT, MPI_COMM_WORLD);
        MPI_Send(&nb_voisins_out[i], 1, MPI_INT, i, TAGINIT, MPI_COMM_WORLD);
        MPI_Send(voisins_in[i], nb_voisins_in[i], MPI_INT, i, TAGINIT,
                 MPI_COMM_WORLD);
        MPI_Send(voisins_out[i], nb_voisins_out[i], MPI_INT, i, TAGINIT,
                 MPI_COMM_WORLD);
    }
}
```

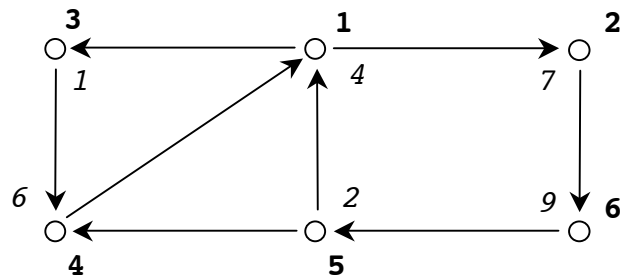


FIGURE 1 – Topologie pour l'exécution de PHASE

```

    MPI_Send(&min_local[i], 1, MPI_INT, i, TAGINIT, MPI_COMM_WORLD);
}
}

```

Le programme principal est identique à celui proposé dans le TME précédent. Les fonctions `main` et `simulateur` sont disponibles sur la page web de l'UE.

### Question 1

Implémentez sous MPI la fonction `calcul_min(rang)` qui réalise un calcul de minimum réparti en appliquant l'algorithme PHASE.

### Question 2

Appliquez votre programme sur la topologie fournie en exemple, en exécutant tous les processus sur des machines différentes.