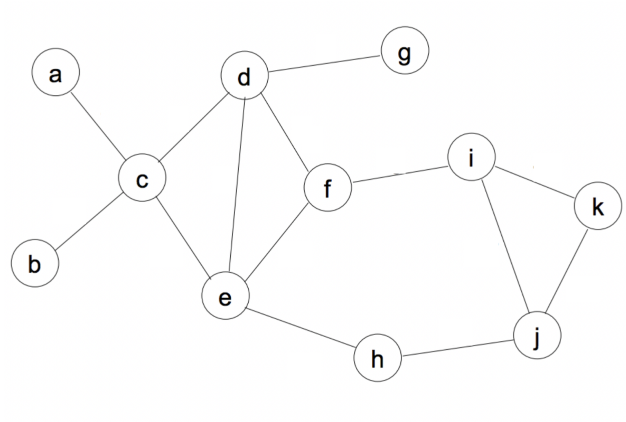
TD2
Algorithmes distribués sur les graphes

La figure suivante sera utilisée dans tout le TD2.



Nous supposerons que le nœud e est la source des algorithmes de construction d'arbre qui sont étudiés par la suite.

Pour analyser la complexité en temps des algorithmes, on supposera qu'un message est émis sur un lien de communication (et reçu par le destinataire de ce lien) en $O(1)$. On négligera, pour cette analyse, les calculs locaux.

1 Exercice 1 - Construction d'un arbre couvrant par inondation

Soit $G = (V, E)$ un graphe connexe. On note $n = |V|$ et $m = |E|$.

On note $d(u, v)$ la distance entre les nœuds u et v du graphe G. $d(u, v)$ correspond au nombre minimum d'arêtes entre u et v dans G.

L'excentricité e d'un nœud u est sa distance maximale à tous les autres sommets dans G : $e(u) = max\{d(u, v), v \in V\}$.

Le rayon R du graphe G est l'excentricité minimum dans G : $R = min\{e(u), u \in V\}$.

Le diamètre D du graphe G est l'excentricité maximum dans G : $R = max\{e(u), u \in V\}$.

Questions :

1. Que valent le rayon et le diamètre du graphe donné dans la figure précédente ?
2. Montrer que $R \leq D \leq 2R$.
3. Supposons que l'algorithme d'inondation est utilisé pour construire un arbre. Dans cet algorithme, la source (e) envoie un message à tous ses voisins et chaque nœud qui reçoit pour la première fois ce message le retransmet à tous ses voisins sauf à celui dont il vient de recevoir le message.

   À partir du graphe exemple, indiquez un ordre d'envoi sur les premiers messages qui ne donnera pas un arbre couvrant en largeur.
4. Quelles sont les complexités, dans le pire cas, en nombre de messages et en temps de l'algorithme d'inondation ?
5. Que faudrait-il rajouter à l'algorithme d'inondation pour que la source puisse savoir que l'arbre a bien été construit ?
6. Quelles sont les complexités, dans le pire cas, en nombre de messages et en temps de l'algorithme proposé à la question précédente ?

2 Exercice - Construction d'un arbre couvrant en largeur

Questions :

1. Indiquez les différents messages échangés lors des deux premières phases de l'algorithme distribué de Zhu & Cheung ("à la Dijkstra") sur le graphe exemple.
2. Quelle est la complexité en nombre de messages échangés pour cet algorithme ?
3. Supposons maintenant que l'algorithme distribué de Cheung ("à la Bellman-Ford") est utilisé sur le graphe exemple. Quelle est la valeur de la distance pour chaque nœud au tout début de l'algorithme ? Quels sont les nœuds qui envoient des messages au tout début de l'algorithme ?
4. Est-ce que, dans le graphe exemple, un nœud peut changer de distance plus de deux fois ?
5. Est-ce que le nœud k peut avoir des parents différents dans deux exécutions différentes de l'algorithme ?
6. Combien de messages sont échangés, dans le pire cas, avec cet algorithme ?

3 Exercice - Coloration et ensemble indépendant maximal

On s'intéresse dans un premier temps à l'algorithme distribué de coloriage glouton vu en cours qu'on va appliquer au graphe exemple donné au début du sujet. On supposera que les IDs des nœuds sont les lettres données dans le graphe exemple et donc que l'ordre lexicographique est utilisé.

Questions :

1. Exécuter cet algorithme sur le graphe exemple. Vous indiquerez le coloriage de chaque nœud à chaque ronde de l'algorithme et vous prendrez les couleurs dans l'ensemble $0; 1; \ldots; max$ où max est le nombre maximum de couleurs dont vous avez besoin pour cet exemple.
2. Quand est-ce qu'un nœud peut arrêter de participer à l'algorithme ?
3. Quel est le nombre de rondes, dans le pire cas, pour cet algorithme ?
4. Quel est le nombre de messages échangés dans le pire cas ?
5. Montrer que, avec cet algorithme, le nombre de couleurs attribuées est inférieur ou égal à $\Delta + 1$ (où Δ est le degré du graphe, c'est-à-dire $max_{v \in V}\{deg(v)\}$) ?

Nous allons maintenant étudier l'algorithme suivant donné dans la figure 1. Attention, dans cet algorithme le paramètre m ne correspond pas au nombre d'arêtes du graphe mais au nombre de couleurs du coloriage initial, constituant le point de départ de l'algorithme.

```
(1)  for each j ∈ neighbors_i do send INIT(color_i[i]) to p_j end for;
(2)  for each j ∈ neighbors_i
(3)          do wait (INIT(col_j) received from p_j); color_i[j] ← col_j
(4)  end for;
(5)  for r_i from (Δ + 2) to m do
       begin asynchronous round
(6)    if (color_i[i] = r_i)
(7)      then c ← smallest color in {1, ..., Δ + 1} such that ∀ j ∈ neighbors_i : color_i[j] ≠ c;
(8)          color_i[i] ← c
(9)    end if;
(10)   for each j ∈ neighbors_i do send COLOR(r_i, color_i[i]) to p_j end for;
(11)   for each j ∈ neighbors_i do
(12)     wait (COLOR(r, col_j) with r = r_i received from p_j);
(13)     color_i[j] ← col_j
(14)   end for
       end asynchronous round
(15) end for.
```

Fig. 2.8 Distributed $(\Delta + 1)$-coloring from an initial m-coloring where $n \geq m \geq \Delta + 2$

FIGURE 1 – Coloriage distribué à partir d'un m-coloriage existant.

7. Appliquer l'algorithme au graphe exemple. On supposera que le coloriage initial est un 11-coloriage et que le nœud a a la couleur 1, le nœud b la couleur 2, ainsi de suite en suivant l'ordre lexicographique.
8. Comparer cet algorithme avec l'algorithme glouton sur le graphe exemple.
9. Combien de rondes a cet algorithme et quel est le nombre de messages échangés ?
10. Comment améliorer cet algorithme sur le nombre de messages échangés ?

Nous allons maintenant étudier l'algorithme distribué de construction d'ensemble indépendant maximal basé sur le coloriage des nœuds (se trouvant dans les transparents du CM3).

12. Exécuter cet algorithme sur le graphe exemple en supposant que le coloriage initial correspond au coloriage obtenu avec l'algorithme de coloriage glouton.
13. Exécuter cet algorithme en utilisant les couleurs initiales dans l'ordre décroissant.
14. Comparer les 2 ensembles indépendants maximaux.
15. Est-ce que ce sont des ensembles indépendants maximum ?