TP n°9 – Arbres couvrants minimaux

1 Un exemple

- **Q1.** Effectuer l'algorithme de Kruskal sur l'exemple suivant et dessiner l'arbre couvrant résultant. Laissé à faire

2 Implémentation de Kruskal

Pour rappel, les fonctions du TP précédent pour unir et trouver :

```
 type unirtrouver = int array;;

let init n =
    let uf = Array.make n 0 in
    for i=0 to n-1 do
        uf.(i)<-i
    done;
    uf;;

let trouver uf i = uf.(i);;

let unir uf i j =
    let ri = uf.(i) in
    let rj = uf.(j) in
    for k = 0 to (Array.length uf) - 1 do
        if uf.(k) = rj then uf.(k)<-ri
    done;;
```

- **Q2.** Implémenter le tri fusion pour une liste de couples. On triera selon l'ordre des <u>premiers éléments</u> des couples.

```
  let rec decoupe l = match l with
    |[]-> [],[]
    |[t] -> [t],[]
    |t::d::q -> let l1,l2=decoupe q in t::l1,d::l2;;

let rec fusion l1 l2 = match l1, l2 with
    |[],_ -> l2
    |_,[] -> l1
    |t1::q1,t2::q2 when t1>=t2 -> t2::(fusion l1 q2)
    |t1::q1,t2::q2 -> t1::(fusion q1 l2);;

let rec tri_fusion l =
    if List.length l < 2 then l
    else
    let l1, l2 = decoupe l in
    fusion (tri_fusion l1) (tri_fusion l2);;
```

- **Q3.** Écrire une fonction qui crée la liste des arêtes et la trie en fonction des poids.

```
  let liste_aretes g =
    let l = ref [] in
    let rec aux aretes s = match aretes with
        |[] -> ()
        |(p,v)::q-> l:=(p,s,v)::!l; aux q s
    in
    for i=0 to Array.length g -1 do
        aux g.(i) i
    done;
    tri_fusion !l;;
```

- **Q4.** En utilisant une implémentation d'union-find (reprendre celle du TP précédent), implémenter l'algorithme de Kruskal

```
  let kruskal g =
    let aretes = liste_aretes g in
    let t = ref [] in (*L'arbre couvrant*)
    let uf = init (Array.length g) in

    let rec aux aretes = match aretes with
        |[] ->()
```

```
        |(p,u,v)::q-> if trouver uf u <> trouver uf v then begin
                              t := (u,v)::!t;
                              unir uf u v
                              end; aux q
    in
    aux aretes; !t;;
```

- **Q5.** Stocker en mémoire le graphe de la partie 1 et tester l'algorithme dessus.

```
  let g1 = [|[(3,2); (6,1)];[(6,0);(2,2);(5,7);(8,4)];[(3,0);(2,1);(1,7);(5,9)];[(10,5)];
             [(8,1);(3,7);(12,6);(8,5)];[(10,3);(4,0);(8,4);(7,8);(20,9)];[(12,7);(12,4);(4,8)];
             [(1,2);(5,1);(3,4);(12,6)];[(4,6);(7,5);(2,9)];[(20,5);(2,8);(5,2)]|];;

kruskal g1;;
```

3 Unicité des arbres couvrants

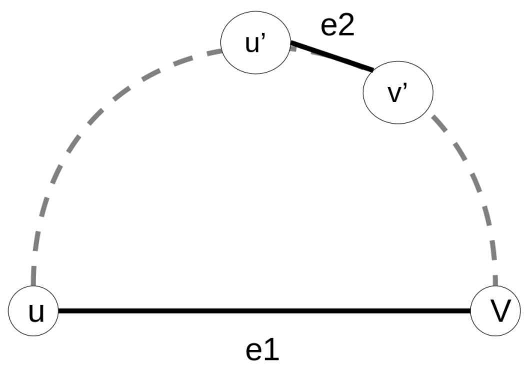
- **Q6.** Trouver un exemple de graphe qui admet plusieurs arbres couvrants minimaux.
  Par exemple un graphe triangle dont tous les poids sont 1.

Soit $G = (S, A, p)$ un graphe non orienté pondéré connexe avec plus de 3 arêtes. On note T un arbre couvrant minimal de G.

On suppose qu'il n'y a pas deux arêtes de même poids. On veut montrer que sous cette hypothèse il y a un seul arbre couvrant minimal pour G.

- **Q7.** Soit T' un arbre couvrant minimal de G. Montrer que $T = T'$.
  Par l'absurde, supposons $T \neq T'$. Alors il existe au moins une arête qui est dans T ou T' mais pas dans l'autre. L'ensemble A des poids des arêtes qui sont dans T ou T' mais pas dans l'autre est donc non vide et fini (car le nombre d'arêtes est fini). Il admet un minimum, noté p, qui est porté par une arête $e_1 = (u, v)$. On supposera $e_1 \in T$, quitte à renommer.

  Dans T', il existe un chemin entre u et v, car T' est connexe. Ce chemin ne contient pas e_1. En revanche il contient $e_2 = (u', v')$ une arête qui n'est pas dans T, car sinon T contient un cycle. Par définition de e_1 et unicité des poids, le poids p' de e_2 est strictement inférieur à p.



  Le graphe $T'' = (T' \setminus \{e_2\}) \cup \{e_1\}$ est un arbre couvrant.

  - Il est acyclique : supposons l'existence d'un cycle, alors ce cycle contient u et v, sinon T' contenait déjà un cycle. Mais si on a un cycle contenant (u, v) dans T'', on a un chemin de u à v dans $T' \setminus \{e_2\}$, donc deux chemins de u à v distincts dans T', i.e. un cycle. C'est absurde.
  - Il est connexe : prenons deux sommets x et y. Dans T', ils sont reliés. On a deux cas de figure : soit le chemin de x à y ne passe pas par e_2, donc existe toujours dans T'', soit il passe par e_2 et est de la forme $x, \ldots, u', v', \ldots, y$. Dans T'' le chemin suivant existe (éventuellement il repasse sur ses pas) $x, \ldots, u', \ldots, u, v, \ldots, v', \ldots, y$ (tous les chemins en pointillés sont ceux qui existent dans T').

  Le poids de T'' est $w(T'') = w(T') - p' + p < w(T')$, ce qui contredit le fait que T' est un arbre couvrant minimal. C'est absurde, $T = T'$.
  Remarque : on aurait aussi pu dire que T'' est un arbre car il est acyclique et contient $|S| - 1$ arêtes. Pour rappel un arbre à n sommets contient toujours $n - 1$ arêtes.

4 Spécificités de certaines arêtes

Supposons d'abord que le poids le plus petit ne soit porté que par une arête : il existe une (et une seule) arête de poids minimal.

- **Q8.** Montrer qu'un arbre couvrant minimal contient nécessairement cette arête. Soit (u, v) la plus petite arête et T un arbre couvrant de poids minimal ne la contenant pas.

  Dans T il existe un chemin entre u et v. Soit e une arête de ce chemin, alors $T' = (T \setminus \{e\}) \cup \{(u, v)\}$ est un arbre couvrant de poids strictement inférieur à T, puisque (u, v) est la seule arête de poids minimum. (c'est le même raisonnement qu'au-dessus, on ne développe pas)

- **Q9.** En rajoutant l'hypothèse que le deuxième poids le plus petit ne soit également porté que par une seule arête, montrer que la deuxième plus petite arête est également dans n'importe quel arbre couvrant minimal.

  Soit T un arbre couvrant minimal. Il contient l'arête de plus petit poids e_1. Soit $e_2 = (u, v)$ l'arête de poids de deuxième plus petit poids. Supposons que T ne contienne pas cette arête.

  Dans T il existe un chemin de (u, v), qui emprunte plus de deux arêtes. Il existe donc une arête e_3 de ce chemin qui n'est pas e_1. $T' = (T \setminus \{e_3\}) \cup \{e_2\}$ est un arbre couvrant de poids strictement inférieur à T, toujours pour les mêmes raisons.

- **Q10.** La troisième plus petite arête (à nouveau on suppose qu'elle est la seule à porter son poids) est elle nécessairement dans tout arbre couvrant de poids minimal ?

  Non, si les trois arêtes les plus petites forment un cycle par exemple, la troisième arête n'appartiendra à aucun arbre couvrant minimal.

En rapport avec la réponse à la question précédente, on peut prouver la propriété suivante :

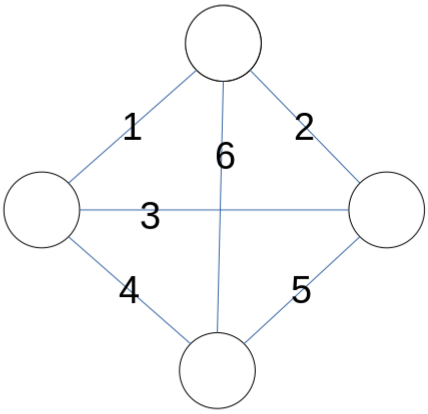
- **Q11.** Montrer que pour chaque cycle dans le graphe, les arbres couvrants minimaux ne contiennent pas l'arête de poids maximal du cycle (à nouveau on suppose qu'elle est unique).

  On suppose l'existence d'un cycle dans G et on note $e_1 = (u, v)$ l'arête de poids maximal de ce cycle. On suppose l'existence de T ACM qui contient cette arête. Il existe au moins une arête e_2 du cycle qui n'est pas dans T.

  $T' = (T \setminus \{e_1\}) \cup \{e_2\}$ est un arbre couvrant de poids strictement inférieur à T, toujours pour les mêmes raisons, ce qui est absurde.

- **Q12.** Un arbre couvrant minimal contient-il l'arête minimale de chaque cycle ?

  Non, dans l'exemple suivant un arbre couvrant minimal est formé par les arêtes de poids 1, 2 et 4. Pourtant les arêtes de poids 3, 4 et 5 forment un cycle dont le poids minimal est 3.



5 L'algorithme de Prim

Non corrigé