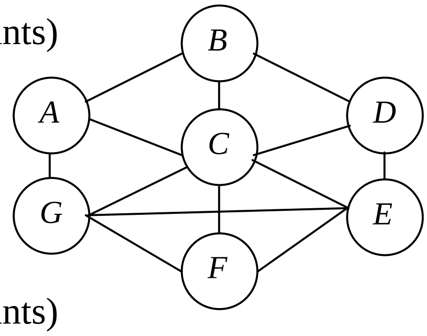
Médian de RO03 - Printemps 2012 - 2heures.

Exercice 1 (4 points)

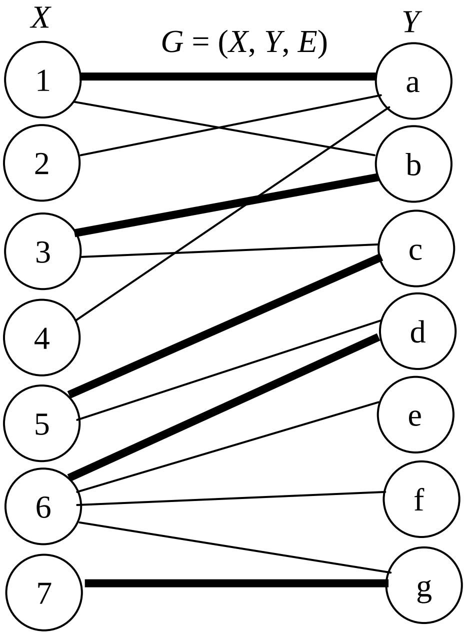


Déterminer pour le graphe G :

(a) un stable maximal ;
(b) un coloriage avec un nombre minimal de couleur ;
(c) une chaîne eulérienne s'il en existe ;
(d) un cycle hamiltonien s'il en existe.

Justifier chaque réponse.

Exercice 2 (4 points)



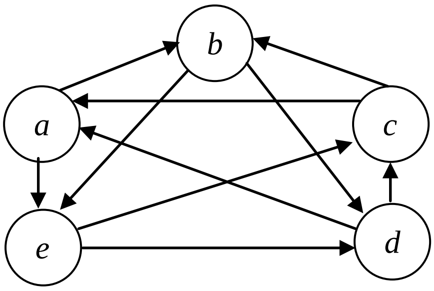
(1) A quelle condition un couplage est-il maximal sur un graphe quelconque ?

(2) Le couplage rapporté sur le graphe biparti $G = (X, Y, E)$ ci-dessus est-il maximal ? Justifier votre réponse : on améliorera le couplage si cela est possible, ou on montrera que le couplage est maximal.

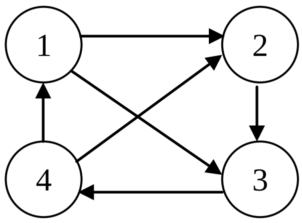
(3) On part du couplage éventuellement amélioré à l'issue de (2), et on utilise la procédure de marquage vu en cours. Rapporter le marquage final M obtenu.

PROBLEME (12 points)

Un tournoi est un graphe orienté tel que chaque paire de sommets est reliée exactement par un arc : pour tout $[i, j]$, on a donc soit l'arc (i, j) soit l'arc (j, i). On va démontrer qu'un tournoi a au moins un chemin hamiltonien. Ces graphes particuliers sont appelés tournoi car ils peuvent modéliser les résultats d'un championnat n'ayant que des matchs « aller ». Le but de ce problème est d'étudier un algorithme qui détermine un chemin hamiltonien dans un « tournoi ».

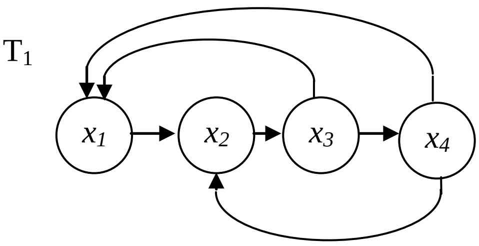


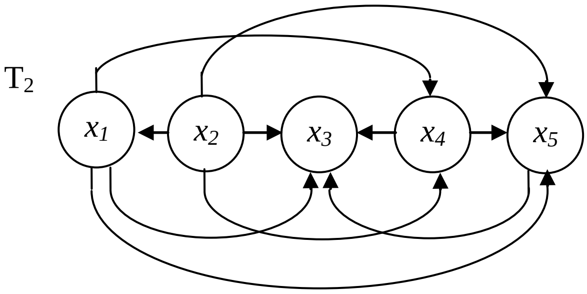
Deux exemples de tournoi :



Question I : Etude d'un algorithme (3 points)

Appliquer l'algorithme ci-dessous aux deux tournois T_1 et T_2. On rapportera les chemins successifs calculés par l'algorithme.





CHEMIN HAMILTONIEN

Input : Un tournoi T sur n sommets étiquetés $x_1, x_2, \ldots, x_n$ et sa matrice d'adjacence $M = [m_{ij}]$.
Output : Un chemin hamiltonien P sur T.

1. $P := x_1$; $i := 2$;
2. Si $i > n$ aller à l'étape 8 ;
3. Soit $P = y_1 y_2 \ldots y_{i-1}$ le chemin courant ;
4. Si $(x_i, y_1) \in T$ alors { $P := x_i P$; Aller à l'étape 7; }
5. *Si* $(y_{i-1}, x_i) \in T$ alors { $P := P\, x_i$; Aller à l'étape 7; }
6. Pour $j := 1$ à $i - 2$ faire
   Si (y_j, x_i) *et* $(x_i, y_{j+1}) \in P$ alors {
   /* on insère x_i dans le chemin entre y_j et y_{j+1} */
   $P := P\,[y_1, y_j]\, x_i\, P\,[y_{j+1}, y_{i-1}]$ et aller en 7 ;
   }
7. $i := i + 1$; aller à l'étape 2 ;
8. Retourner le chemin P ;

Question II : Complexité de l'algorithme (2 points)

On suppose que la matrice M est déjà en mémoire, c'est-à-dire qu'il n'y a pas besoin d'écrire M. Quelle structure de données utilisez-vous pour coder le chemin P ?
Quelle est la complexité résultante de l'algorithme ? Justifier.

Question III : Application au tri (1 point)

On veut trier n nombres distincts en utilisant l'algorithme précédent. Proposer une méthode et évaluer sa complexité. Quelle est la technique de tri correspondante. A-t-on besoin d'écrire la matrice M ?

Question IV : Preuve de l'algorithme (3 points)

1) Justifier la méthode d'insertion de x_i de l'algorithme permettant de construire un chemin de longueur i à partir d'un chemin de longueur i-1.

2) Montrer la validité de l'algorithme.

Question V : Questions subsidiaires (3 points)

L'algorithme ci-dessous, plus général que l'insertion d'un sommet dans un chemin, a pour objet de fusionner deux chemins P et P'.

FUSION(P, P')
Input : deux chemins disjoints $P = x_1\, x_2 \ldots x_k$ et $P' = y_1\, y_2 \ldots y_r$.
Output : Un chemin P^* passant par les $k+r$ sommets.

1. Si P' est vide, alors $P^* := P$;
2. Si P est vide, alors $P^* := P'$;
3. Si x_1 précède y_1, alors $P^* := x_1$ FUSION(P-x_1, P') ;
4. Si y_1 précède x_1, alors $P^* := y_1$ FUSION(P, P'-y_1) ;
5. Retourner P^* ;

1) Quelle est la complexité de FUSION en fonction de k et r ?

2) Un algorithme de tri en $O(n \log n)$ consiste à trier les nombres par paquets de 2, de 4, de 8, *etc*… Peut-on en déduire un algorithme plus efficace pour la recherche d'un chemin hamiltonien d'un tournoi ? Expliquer la « partie centrale » de l'algorithme.