Final IT4A

Prénom, NOM :

PARTIE 2 : deuxième copie

Exercice 3. *Affectation de développeurs à des projets* Une entreprise de services informatiques doit affecter des développeurs à trois projets P_1, P_2, P_3. Les effectifs disponibles sont les suivants :

Type de développeur	Effectif
J (junior)	3
C (confirmé)	2
E (expert)	2

Chaque développeur est affecté à *un seul* projet. Les projets ont une *capacité d'accueil* limitée (nombre de postes ouverts) :

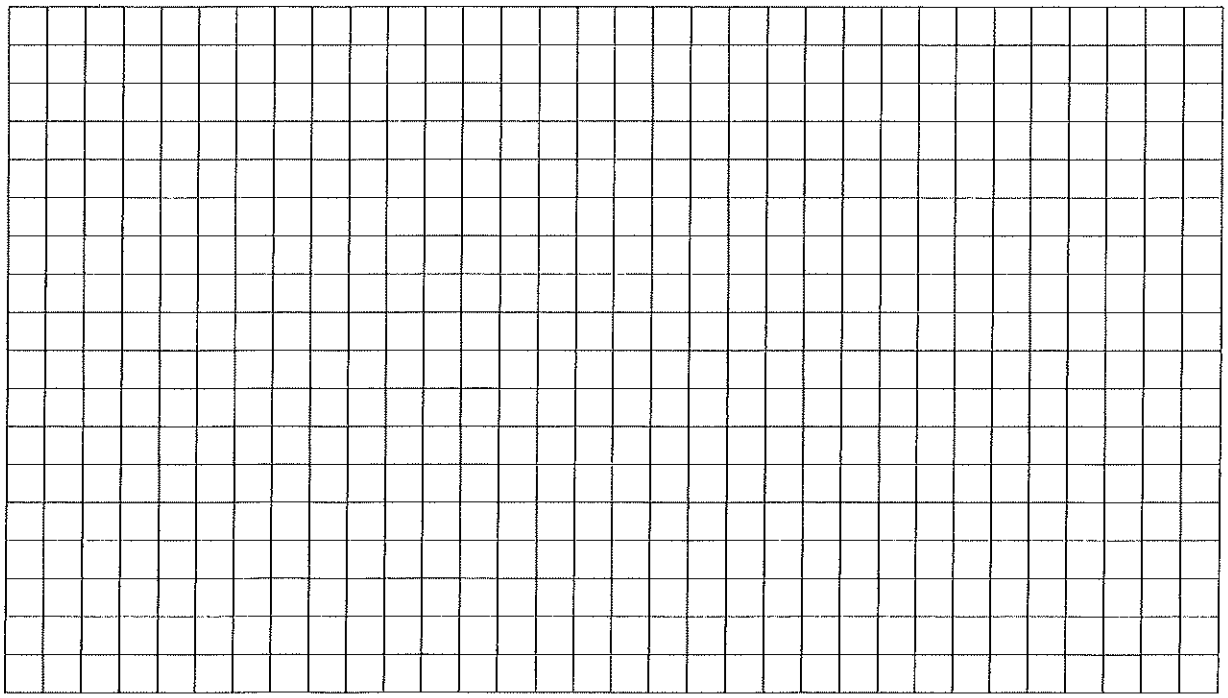
Projet	Capacité d'accueil
P_1	2
P_2	2
P_3	3

Enfin, les compétences requises limitent les affectations possibles :

Type	Projets accessibles
J (junior)	P_1, P_2
C (confirmé)	P_1, P_3
E (expert)	P_1, P_3

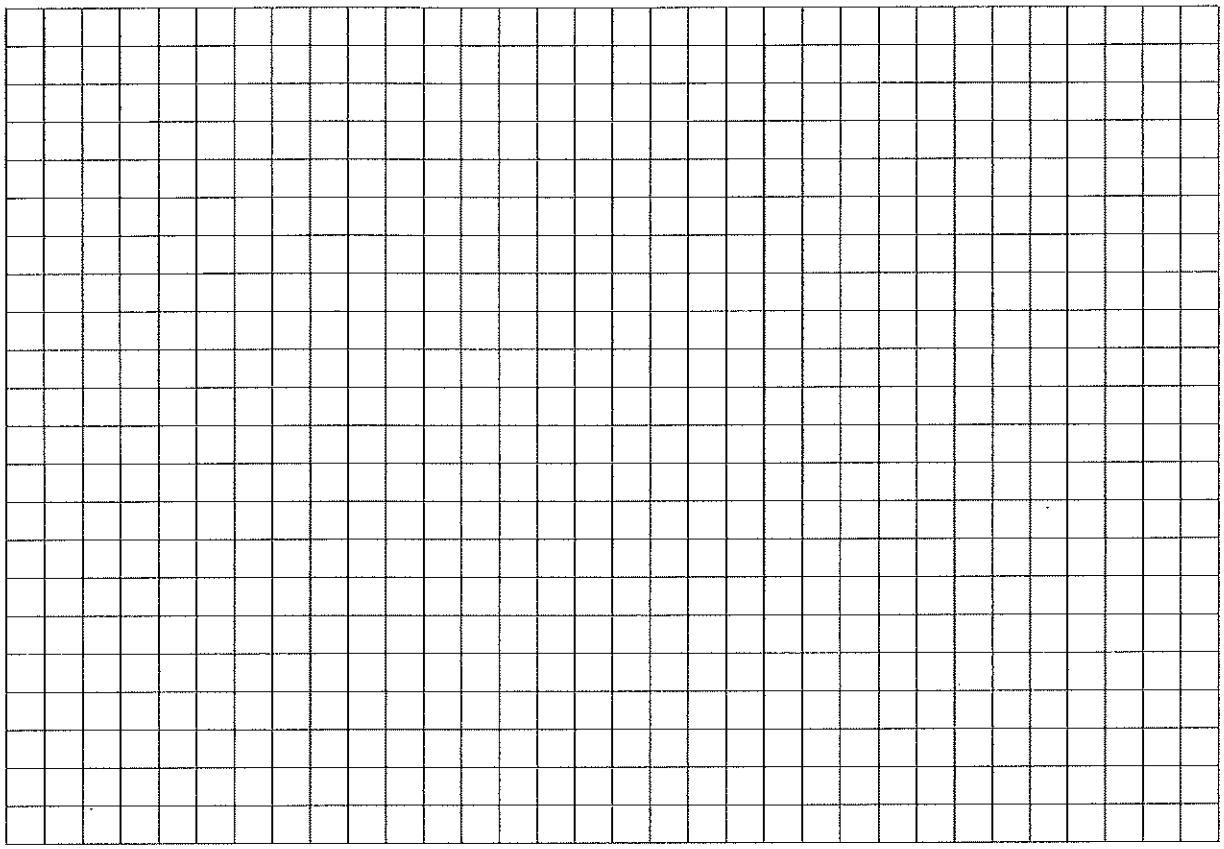
Par règle de gestion, **au plus 2 développeurs d'un même type** peuvent être affectés à un même projet.

1. **Modélisation.** Construire un réseau de transport $R = (S, A, s, p, c)$, avec $S = \{s, J, C, E, P_1, P_2, P_3, p\}$, modélisant ce problème : on précisera les sommets, les arcs et les capacités.

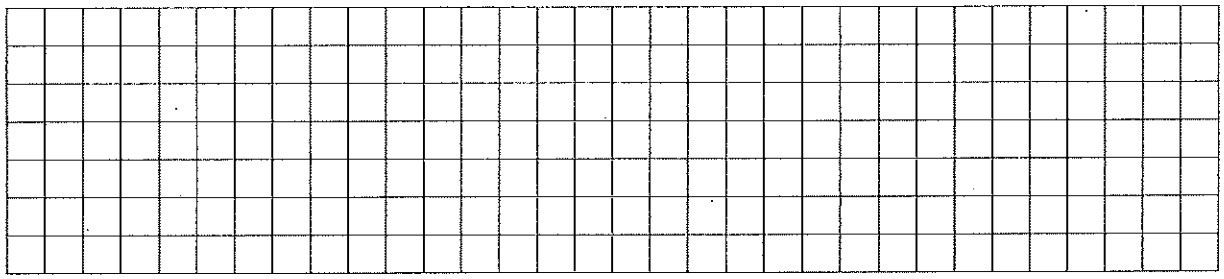


2. **Flot maximal.** Appliquer l'algorithme de Ford–Fulkerson au réseau R, **en partant du flot nul**, avec la règle de choix suivante : lors de la recherche d'un chemin ou d'une chaîne augmentante, on explore toujours le sommet le *plus petit* parmi les sommets *atteignables* et *non visités* : on prendra l'ordre $s <$

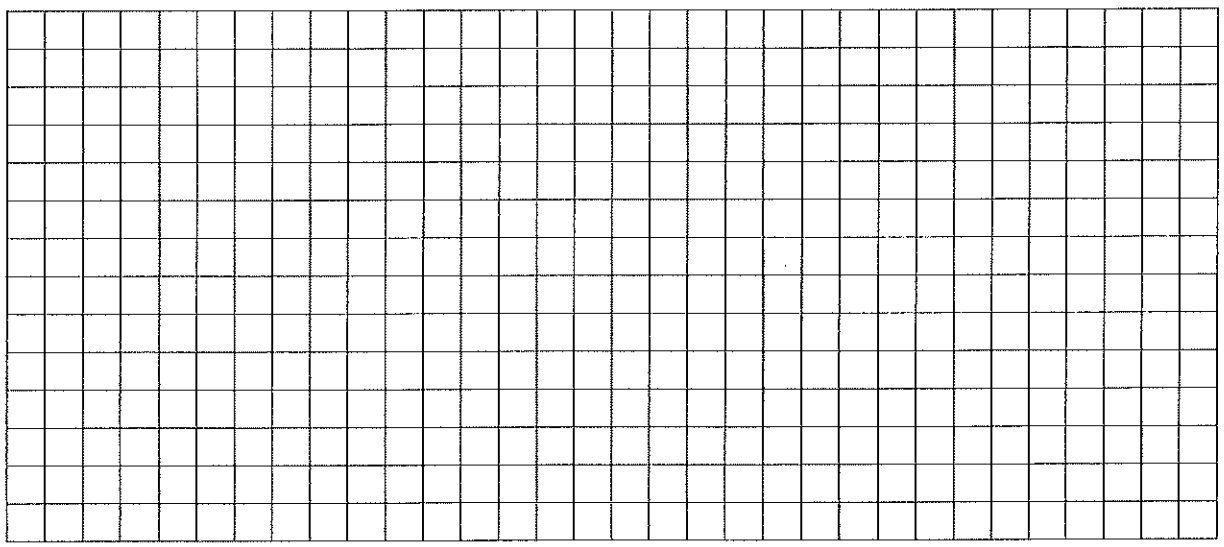
$J < C < E < P_1 < P_2 < P_3 < p$, les arcs retour du réseau résiduel étant utilisés dès qu'ils mènent au plus petit sommet. Détailler chaque augmentation (chemin ou chaîne, valeur de δ, arcs retour éventuels), justifier l'arrêt, et donner la valeur du flot maximal.



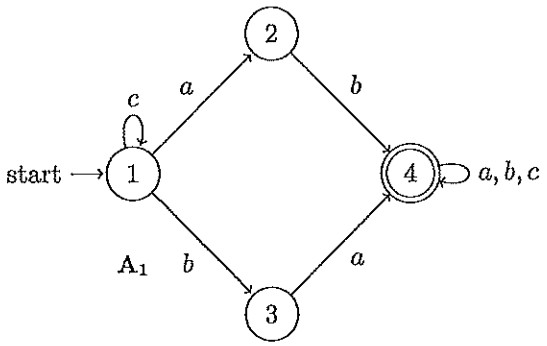
3. **Coupe.** Exhiber une coupe de capacité minimale et vérifier l'égalité flot maximal / coupe minimale.

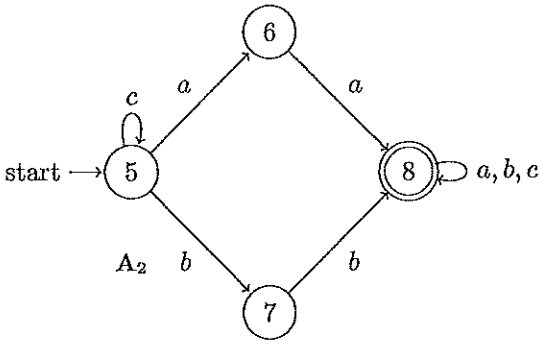


4. **Variante.** L'entreprise recrute un confirmé supplémentaire (le type C passe à 3 disponibles) et le projet P_3 ouvre un poste supplémentaire (capacité d'accueil 4). Modifier le réseau, puis déterminer la nouvelle valeur du flot maximal (on justifiera par un flot et une coupe).



Exercice 4 (*Union, déterminisation et simplification d'automates).* L'alphabet est $\Sigma = a \mid b \mid c$. On considère les deux automates ci-dessous. Pour chacun, 1 (resp. 5) est l'état de départ (flèche entrante) et 4 (resp. 8) est l'état final (double cercle).





1. L'automate A_1 est-il déterministe ? Est-il complet ? Décrire le langage $L_1 = L(\mathrm{A}_1)$ reconnu par A_1 (on justifiera les réponses).

2. Mêmes questions pour A_2 : est-il déterministe, complet, et quel langage $L_2 = L(\mathrm{A}_2)$ reconnaît-il ?

3. On définit $\mathrm{A} = \mathrm{A}_1 \mid \mathrm{A}_2$ comme l'union des deux automates : états $E = 1 \mid \ldots \mid 8$, états de départ $1 \mid 5$, états finaux $4 \mid 8$, toutes les transitions de A_1 et de A_2.
   L'automate A est-il déterministe ? Justifier. Donner une expression régulière du langage $L = L(\mathrm{A})$.

4. Déterminiser A à l'aide de l'algorithme de *subset construction*. Vous obtiendrez un automate à 6 états en comptant l'état de `blocage` (l'état "poubelle"). Dessinez l'automate obtenu (ne pas mettre l'état "poubelle").

5. Justifier que les états 4 et 8 peuvent être fusionnés en un unique état. Dessiner l'automate simplifié obtenu.

6. Cet automate simplifié peut-il être simplifié encore ? Justifier la réponse.

7. Donner une expression régulière du langage reconnu par l'automate simplifié, et la comparer avec l'expression obtenue à la question 3. Pouvais-t-on prédire cette expression à l'avance ?