Licence Informatique et GMI — Algorithmique et ROAD — Septembre 2006
Durée 3 heures — documents et calculatrices interdits

Prenez le temps de faire les calculs au brouillon avant de les recopier sur votre copie d'examen. Efforcez–vous de traiter les questions dans l'ordre. Indiquez bien le numéro de chaque question avant la réponse. Les copies mal présentées seront pénalisées.

1 Modélisation (13 points)

Une entreprise d'interim en informatique comporte 300 salariés. Parmi ces salariés, certains vont suivre un stage dans l'année à venir, les autres non. Ces stages sont de natures différentes : bureautique, Anglais et administration des réseaux. Il y a, pour chaque stage, un nombre minimal et un nombre maximal de stagiaires possibles. Chacun de ces stages a bien sûr un coût, proportionnel au nombre de salariés inscrits.

Chaque année, des salariés quittent l'entreprise. En se basant sur l'expérience des années précédentes, l'entreprise estime que, parmi les salariés qui ne suivront pas de stage, sept personnes sur dix resteront dans l'entreprise à la fin de l'année. Ce rapport de $7/10$ est appelé le *taux de fidélité* des salariés non inscrits en stage. Des taux de fidélité pour les salariés inscrits en stage ont pu être estimés. Ils sont bien sûr plus élevés que celui des salariés non inscrits en stage. Le tableau suivant synthétise les différentes données disponibles.

	fidélité	coût (euros/salarié)	nb. minimum inscrits	nb. maximum inscrits
bureautique	.80	500	20	100
Anglais	.85	400	20	40
réseaux	.95	1000	5	20

Chaque personne quittant l'entreprise à la fin de l'année devra être remplacée par au moins un nouveau salarié, de telle sorte que l'effectif à la fin de l'année soit supérieur ou égal à l'effectif actuel. L'entreprise souhaite qu'au moins trente et au plus soixante dix nouvelles personnes soient embauchées. Toutefois, chaque nouveau salarié doit subir une première formation avant d'être opérationnel. Le coût de cette première formation est estimé à 3500 euros par salarié.

On souhaite déterminer le nombre d'anciens salariés à inscrire dans les différents types de stages offerts ainsi que le nombre de nouveaux salariés à embaucher de telle sorte que le coût global (coûts du stage pour les anciens plus coût de première formation pour les nouveaux) soit minimal.

Question 1 [3 pts]. Compléter le modèle AMPL ci–dessous, destiné à résoudre ce problème. Utiliser toutes les variables définies.

SOLUTION. Voici les contraintes et l'objectif manquants.

```
minimize frais :
    embauches * cout_formation + sum {s in STAGES} inscrits [s] * cout [s];

subject to calcule_non_inscrits :
    non_inscrits + sum {s in STAGES} inscrits [s] = effectif;

subject to maintien_effectif :
    embauches +
        fidelite_non_inscrits * non_inscrits +
        sum {s in STAGES} fidelite [s] * inscrits [s] >= effectif;

set STAGES;
# Entreprise
param effectif;                    # salariés
# Nouveaux salariés
param cout_formation;              # euros / salarié
param min_embauches;               # salariés
param max_embauches;               # salariés
var embauches >= min_embauches, <= max_embauches;   # salariés
# Anciens salariés en stage
param fidelite {STAGES};           # sans dimension, entre 0 et 1
param cout {STAGES};               # euros / salarié
param inscrits_min {STAGES};       # salariés (nb min inscrits)
param inscrits_max {STAGES};       # salariés (nb max inscrits)
var inscrits {STAGES} >= 0;        # salariés
# Anciens salariés non inscrits en stage
param fidelite_non_inscrits;       # sans dimension, entre 0 et 1
var non_inscrits >= 0;             # salariés

subject to nbre_inscr {s in STAGES} :
    inscrits_min [s] <= inscrits [s] <= inscrits_max [s];
```

Question 2 [2 pts]. Compléter la partie « données » ci–dessous.

SOLUTION. Voici les parties manquantes.

```
param :       fidelite cout inscrits_min inscrits_max :=
bureautique      .80     500     20            100
anglais          .85     400     20             40
reseaux          .95    1000      5             20;

set STAGES := bureautique anglais reseaux;
param effectif := 300;
param min_embauches := 30;
param max_embauches := 70;
param cout_formation := 3500;
param fidelite_non_inscrits := .70;
```

Question 3 [2 pts]. Que peut–on déduire des valeurs affichées par les commandes suivantes ?

SOLUTION. L'objectif réalisé à l'optimum est de 326000 euros. Il y a 40 inscrits en Anglais, 90 inscrits en bureautique et 20 en réseaux. Les contraintes sur les effectifs pour la bureautique sont inactives. Les autres sont actives.

```
ampl: solve;
MINOS 5.5: optimal solution found.
5 iterations, objective 326000
ampl: display nbre_inscr.lb, nbre_inscr.body, nbre_inscr.ub;
:               nbre_inscr.lb nbre_inscr.body nbre_inscr.ub    :=
anglais                20            40            40
bureautique            20            90           100
reseaux                 5            20            20
```

Question 4 [3 pts]. Comment appelle–t–on les nombres affichés par la commande suivante ? Quelle est leur dimension ? En donner une interprétation.

> SOLUTION. Ce sont les valeurs marginales de la contrainte *nbre_inscr*. Elles ont la dimension d'euros par salariés. Interprétation : si on autorisait une personne de plus à s'inscrire en Anglais, les frais diminueraient de 350 euros ; si on autorisait une personne de plus à s'inscrire en réseaux, les frais diminueraient de 250 euros. Par contre, une (petite) modification sur les nombres maximal et minimal d'inscrits pour la bureautique n'aurait aucune incidence sur les frais (contraintes inactives).

```
ampl: display nbre_inscr;
nbre_inscr [*] :=
    anglais  -350
bureautique     0
    reseaux  -250
```

1.1 Variante

L'entreprise souhaite changer sa façon de gérer les stages. À partir de maintenant, pour chaque stage, deux possibilités se présentent : *soit le stage ouvre* et dans ce cas, le nombre de salariés inscrits doit être compris entre le nombre minimal et le nombre maximal comme précédemment, *soit le stage n'ouvre pas* et dans ce cas, aucun salarié n'y est inscrit.

On souhaite modéliser cette variante du problème par un programme linéaire en *variables entières*. Pour cela, on déclare une variable *ouvert*, indicée par les stages et ne pouvant prendre que les valeurs 0 ou 1. Cette variable s'ajoute aux déclarations précédentes.

```
var ouvert {STAGES} binary;
```

On procède ainsi : les réponses aux deux premières questions ci–dessous contraignent la variable *ouvert [s]* à valoir 0 si *inscrits [s]* vaut zéro et à valoir 1 si *inscrits [s]* est différent de zéro (l'indice s représente un élément de *STAGES*). Cela étant fait, la réponse à la troisième question impose que la variable *inscrits [s]* soit supérieure ou égale à *inscrits_min [s]* si elle est non nulle.

Question 5 [1 pt]. Écrire une contrainte AMPL linéaire qui impose :

$$\textit{inscrits [s]} = 0 \quad \Rightarrow \quad \textit{ouvert [s]} = 0.$$

> SOLUTION. Voici la contrainte.

```
subject to Q5 {s in STAGES} : inscrits [s] >= ouvert [s];
```

Question 6 [1 pt]. Écrire une contrainte AMPL linéaire qui impose (*indication : la variable* inscrits [s] *est nécessairement inférieure ou égale à l'effectif de l'entreprise*) :

$$\textit{inscrits [s]} > 0 \quad \Rightarrow \quad \textit{ouvert [s]} = 1.$$

SOLUTION. Voici la contrainte.

```
subject to Q6 {s in STAGES} : inscrits [s] <= ouvert [s] * effectif;
```

Question 7 [1 pt]. Écrire une contrainte AMPL linéaire qui impose :

$$\textit{ouvert [s]} = 1 \quad \Rightarrow \quad \textit{inscrits [s]} \geq \textit{inscrits\_min [s]}.$$

SOLUTION. Voici la contrainte. Remarque : cette contrainte rend inutile la contrainte de la question 5 ; il resterait encore à modifier la contrainte *nbre_inscr*.

```
subject to Q7 {s in STAGES} : inscrits [s] >= inscrits_min [s] * ouvert [s];
```

2 Le simplexe (7 points)

On considère le programme linéaire suivant.

$$\left\{\begin{array}{rcl} x_1 + 2\,x_2 & = & z[\max] \\ x_1 + x_2 & \leq & 5 \\ x_1 & \leq & 3 \\ x_1 - 2\,x_2 & \geq & 4 \\ x_1, x_2 & \geq & 0 \end{array}\right.$$

Question 8 [2 pts]. Résoudre graphiquement ce programme.

SOLUTION. Programme sans solution.

Question 9 [1 pt]. Ce programme pose–t–il un problème de démarrage ? Justifier.

SOLUTION. Oui. L'origine n'appartient pas au domaine des solutions réalisables (contrainte 3).

Question 10 [3 pts]. Résoudre ce programme par la méthode des deux phases (écrire le programme artificiel et le résoudre).

SOLUTION. Voici les itérations. On voit que le programme est infaisable au fait que l'objectif réalisé à la fin de la première phase est non nul.

```
linear program under standard form
x1 + 2*x2 = z[max]
x1 + x2 + x3 = 5
x1 + x4 = 3
x1 - 2*x2 - x5 = 4
x1, x2, x3, x4, x5 >= 0
```

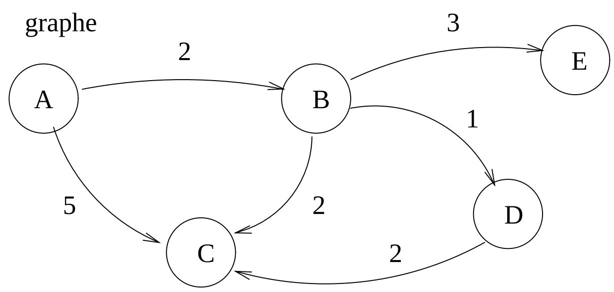
```
slack variables = [x3, x4, x5]
After resolution ...
take the opposite value of the objective

------------------------------------------------------------

PHASE I. RESOLUTION OF THE ARTIFICIAL PROGRAM:

linear program equivalent to a tableau
x1 - 2*x2 - x5 - 4 = w[max]
x1 + x2 + x3 = 5
x1 + x4 = 3
x1 - 2*x2 - x5 + x6 = 4
x1, x2, x3, x4, x5, x6 >= 0
slack variables = [x3, x4, x5]
artificial variables = [x6]

------------------------------------------------------------

PHASE I. APPLICATION OF THE DANTZIG ALGORITHM

    x1  x2  x3  x4  x5  x6
   ------------------------------
x3   1   1   1   0   0   0  |  5
x4   1   0   0   1   0   0  |  3
x6   1  -2   0   0  -1   1  |  4
   ------------------------------
     1  -2   0   0  -1   0  | 4 = - w0

    x1  x2  x3  x4  x5  x6
   ------------------------------
x3   0   1   1  -1   0   0  |  2
x1   1   0   0   1   0   0  |  3
x6   0  -2   0  -1  -1   1  |  1
   ------------------------------
     0  -2   0  -1  -1   0  | 1 = - w0

Infeasible linear pgm
```

Question 11 [1 pt]. Si on résolvait le dual de ce programme, quel(s) type(s) de résultat(s) pourrait–on obtenir (on ne demande ni d'écrire le dual ni de le résoudre) ?

SOLUTION. Un programme linéaire infaisable ou non borné (cf. poly, thm 4). Note : le programme dual est en fait non borné.

3 Graphes (5 points)

On considère le graphe suivant.



Question 12 [1 pt]. Donner sa matrice d'adjacence (ne pas tenir compte des valeurs des arcs).

SOLUTION. Voici la matrice. Les lignes et les colonnes sont numérotées suivant l'ordre alphabétique.

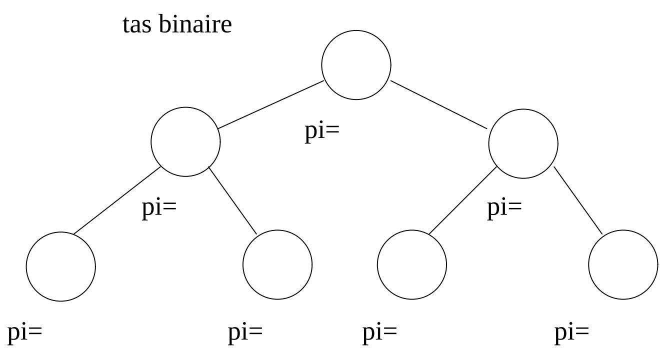
$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

On cherche à calculer la valeur minimale des chemins de A vers tous les autres sommets.

Question 13 [1 pt]. Peut–on employer l'algorithme de Bellman ? Justifier en quelques mots.

SOLUTION. Oui. C'est un graphe acyclique.

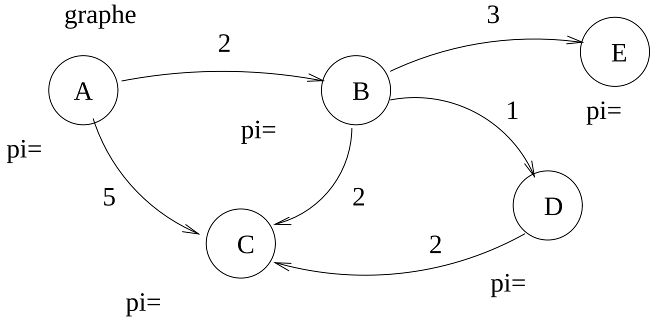
Question 14 [3 pts]. Appliquer l'algorithme de Dijkstra, en gérant la file avec priorité au moyen d'un tas binaire. Au début de chaque itération, donner une configuration possible du tas en utilisant la représentation graphique ci–dessous. Indiquer la couleur de chaque sommet. Indiquer la valeur courante *pi* pour chaque sommet présent dans le tas et pour chaque sommet du graphe pour lequel cette information est pertinente. Il est fortement recommandé d'utiliser la feuille préremplie jointe à l'énoncé. On rappelle les couleurs à utiliser : *rouge* (traitement fini), *vert* (traitement commencé mais pas fini) et *bleu* (traitement pas encore commencé).



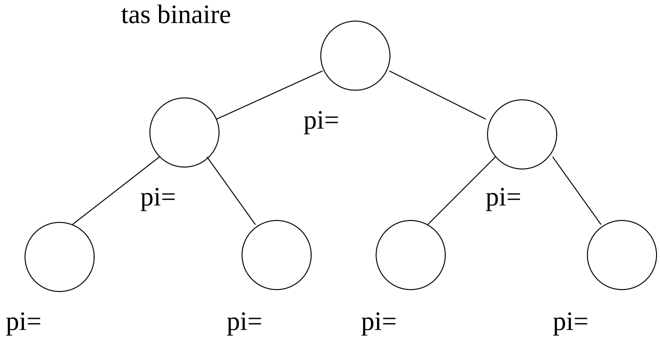
Annexe pour la question sur l’algorithme de Dijkstra

Numéro de place (seul indice de recherche en cas de perte) :

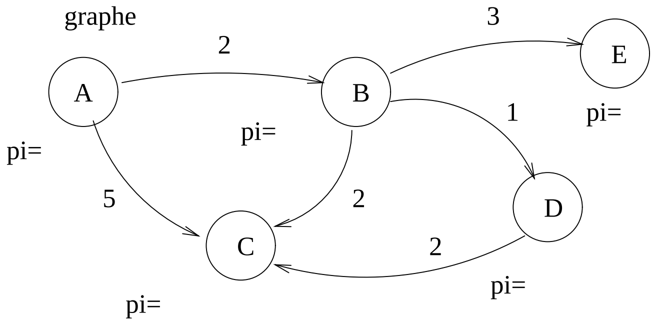
graphe



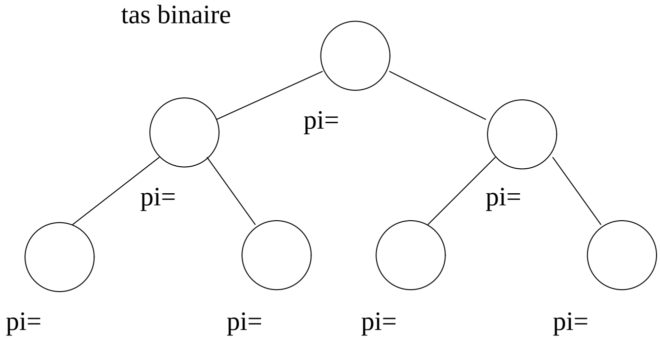
tas binaire



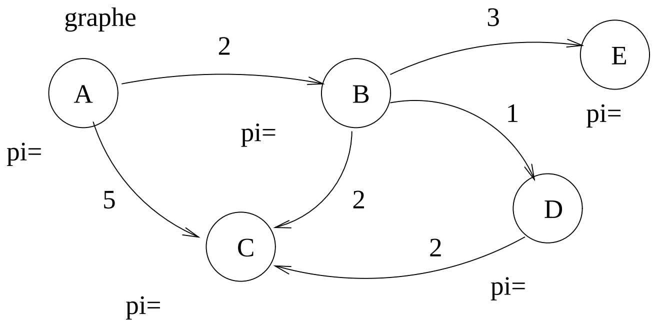
graphe



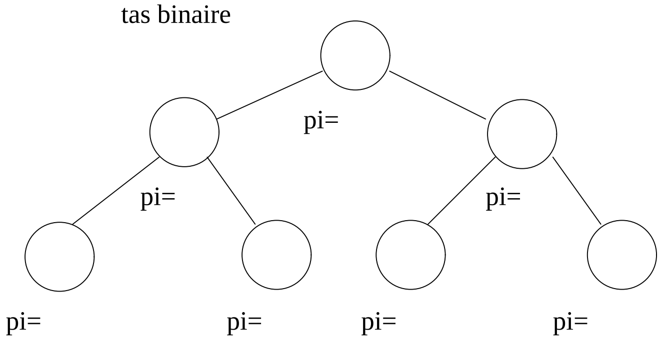
tas binaire



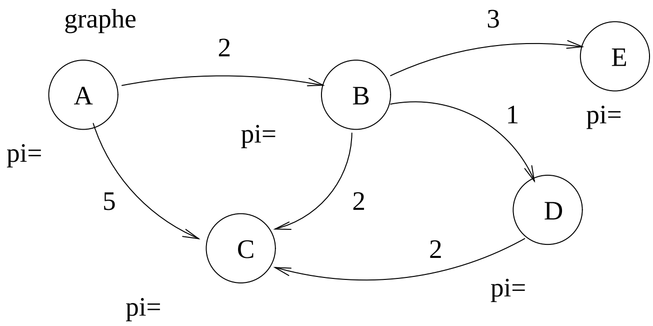
graphe



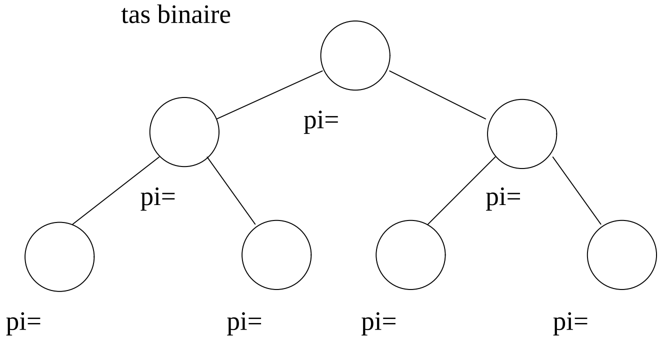
tas binaire

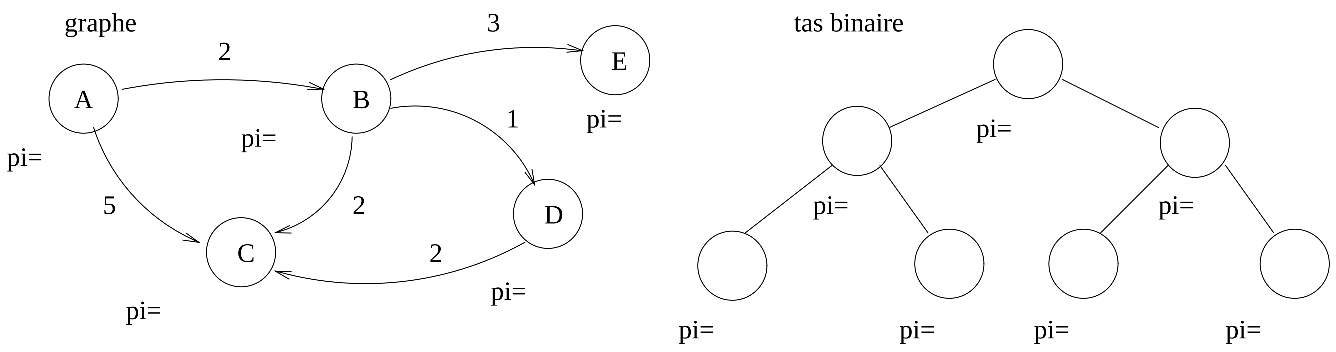


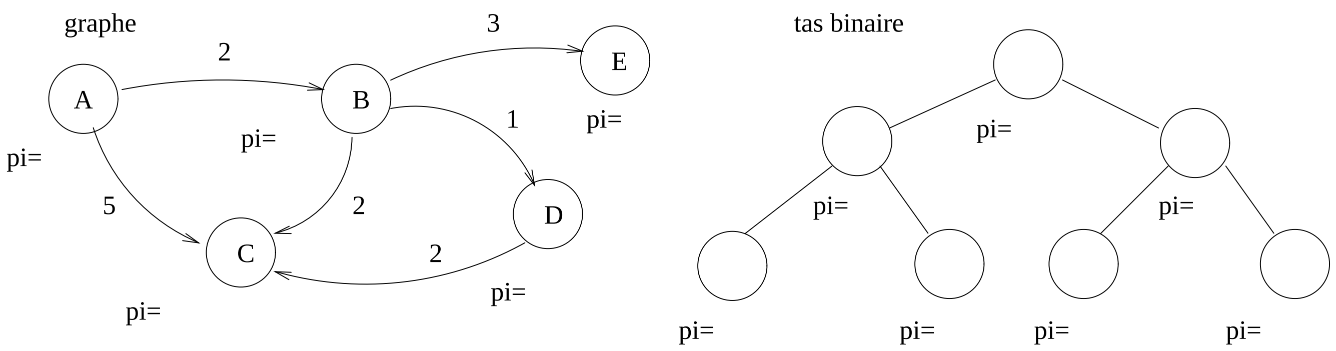
graphe

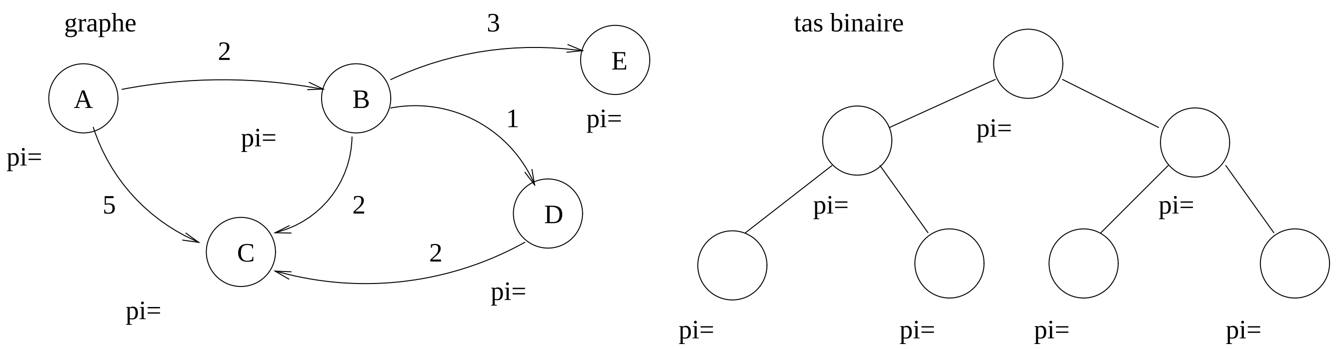


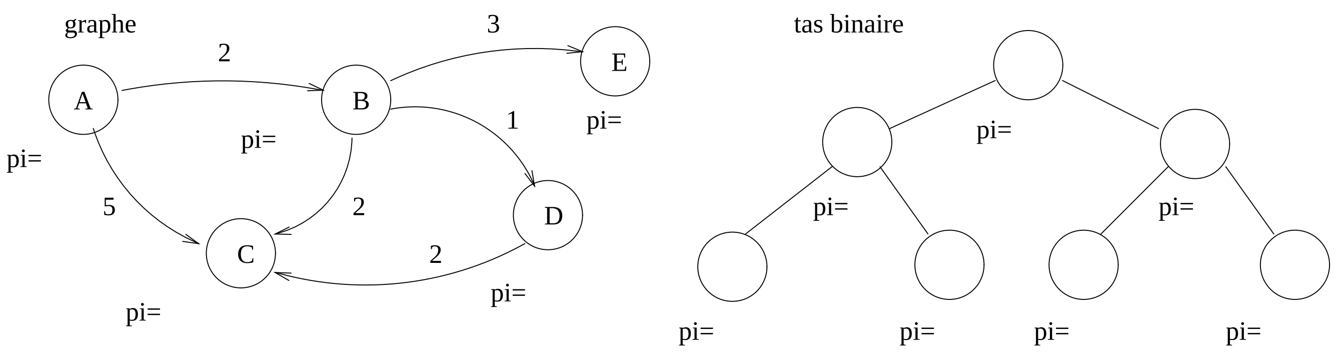
tas binaire




graphe
2
3
A
B
E
pi=
1
pi=
pi=
5
2
D
C
2
pi=
pi=
tas binaire
pi=
pi=
pi=
pi=
pi=
pi=
pi=


graphe
2
3
A
B
E
pi=
1
pi=
pi=
5
2
D
C
2
pi=
pi=
tas binaire
pi=
pi=
pi=
pi=
pi=
pi=
pi=


graphe
2
3
A
B
E
pi=
1
pi=
pi=
5
2
D
C
2
pi=
pi=
tas binaire
pi=
pi=
pi=
pi=
pi=
pi=
pi=


graphe
2
3
A
B
E
pi=
1
pi=
pi=
5
2
D
C
2
pi=
pi=
tas binaire
pi=
pi=
pi=
pi=
pi=
pi=
pi=