

1) def split stack (P1):

Je ne sais pas écrire ce programme avec 2 piles et 1 file.
(On a vu en TP comment le faire avec 3 piles, pas avec des files)

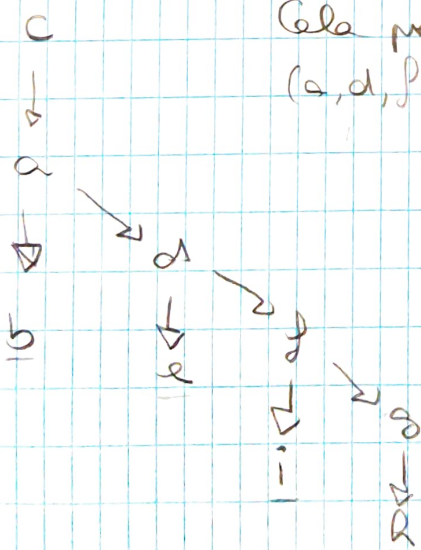
2) 1) Oui car c'est un graphe connexe (on peut aller de n'importe quel sommet à l'autre).

2) Non, car il n'existe pas de parcours Eulérien

3) Non car il existe des de parcours Hamiltonien différents.

4) Il y a 9 chemins \forall pour aller de a vers a en 7 arêtes (liens hypertexte)

5)



Cela prouve bien que c'est un graphe connexe
(a, d, f, g) donne avec ces 2 1 site.

(pas au programme.)

3) Arbre qui a comme particularité que fils droit > racine > fils gauche

b) def is-ABR(a):

if a is None:

return True

if (fils-g(a) is not None and n <= fils-g(a)):

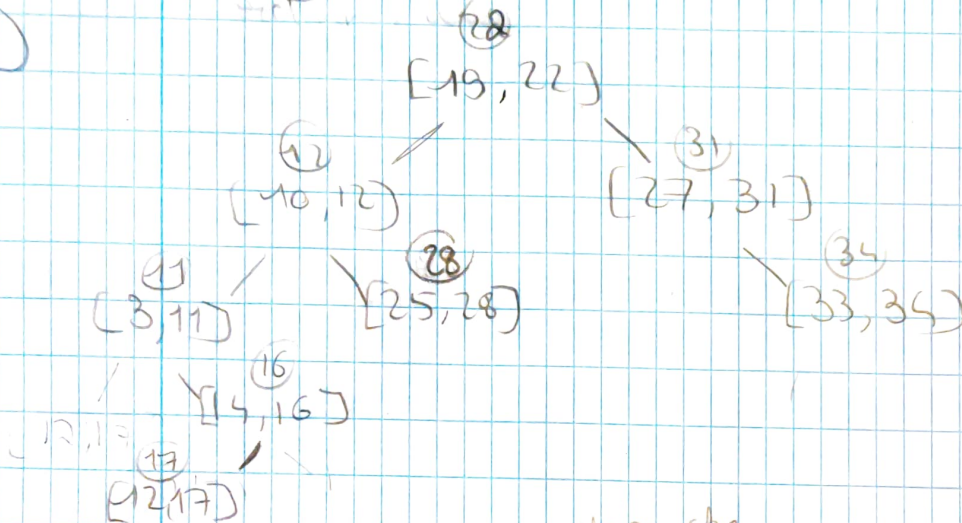
return False

if (fils-d(a) is not None and n >= fils-d(a)):

return False

return (is-ABR(fils-g(a)) and is-ABR(fils-d(a)))

2)



2.2)

Non car le sous arbre du nœud [10, 12] a + d'1 de hauteur que celui de droite

2.3) Def inf(a):

if a is None:

return SSSSS

if a.empty-Tree():

return a.racine()

else:

return min(min(a.racine()), inf(a.chauche()), min(a.racine()), inf(a.droite()))

2.4) Non car val > max(fils gauche), donc val n'est pas dans fils gauche

b) Je ne suis pas du tout sûr que la c

c)

Q.3)

2) Les AI ne prennent pas en compte la borne sup
donc par exemple, on peut poser t_1 si borne sup(t_1)
est inférieure ou supérieure par rapport à borne inf t_2

2)

3)

[Pour ces dernières questions de cet essai, j'ai vraiment
essayé de le faire en m'aidant d'internet, du cours
et des TP mais je n'ai rien trouvé