

Leçon III.1 – Examen final 2016 exercice 2

En supposant que tous les registres sont à zéro au départ, que contient le registre $r2$ à la fin du programme ci-dessous.

1: soustr $r1, \leftarrow 5, -r2$

2: soustr $r2, 7, r3$

3: soustr $r3, 26, r3$

4: soustr $r1, 0, r1$

5: soustr $r2, r2, r1$

6: cont_pp $r2, r3, 5$

r_1	r_2	r_3
5	0	0
5	7	0
5	7	26
-5	7	26
-5	12	26
	17	
	22	
	27	✓

$$r_1 \leftarrow 5 - r_2$$

$$\left. \begin{array}{l} r_2 < r_3 \\ r_2 > r_3 \end{array} \right\}$$

Leçon III.1 – Examen final 2015 question 11

L'instruction « charge_adr r1, r2 » met dans le registre r1 la valeur stockée en mémoire à l'adresse contenue dans r2 (pointeur). Par exemple si r2 contient la valeur 123, cette instruction mettra dans r1 la valeur stockée en mémoire à l'adresse 123.

L'instruction « escrit_adr r1, r2 » écrit en mémoire à l'adresse contenue dans r2 la valeur stockée dans r1. Par exemple si r2 contient la valeur 123 et r1 la valeur 45, cette instruction écrira la valeur 45 à l'adresse 123 en mémoire.

Si l'on suppose que l'on exécute toujours ce programme avec une valeur de r1 strictement supérieure à celle de r0, et si l'on note n la différence entre la valeur contenue dans r1 et celle de r0, **quelle est** alors en fonction de n **la complexité** du programme assembleur suivant :

```
1: charge r2, r0
2: somme r3, r2, 1
3: charge_adr r4, r2
4: charge_adr r5, r3
5: continue_ppe r4, r5, 9
6: escrit_adr r4, r3
7: escrit_adr r5, r2
8: continue 1
9: charge r2, r3
10: continue_ppe r2, r1, 2
```

$$R_1 = \neq R_2$$

$$R_1 \rightarrow R_0$$

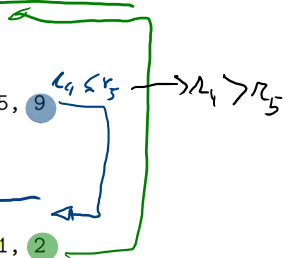
$$f(R_0, R_1)$$
$$n \leftarrow R_1 - R_0$$

Leçon III.1 – Examen final 2015 question 11

$f(n_0, n_1) \quad m \leftarrow n_1 - n_0$

```

1: charge r2, r0
2: somme r3, r2, 1
3: charge_adr r4, r2
4: charge_adr r5, r3
5: continue_ppe r4, r5, 9
6: ecrit_adr r4, r3
7: ecrit_adr r5, r2
8: continue 1
9: charge r2, r3
10: continue_ppe r2, r1, 2
    
```



```

n2 ← n0
do
  n3 ← n2 + 1
  n4 ← *n2
  n5 ← *n3
  if (n4 > n5)
    Swap(*n2, *n3)
    f(n0, n1)
  else
    r2 ← n3
while (n2 ∈ n1)
    
```

Leçon III.1 – Examen final 2016 exercice 1

Considérons le code assembleur suivant :

```
1: cont_egal r1, r2, 7  
2: cont_pp r1, r2, 5  
3: soustrait r1, r1, r2  
4: continue 1  
5: soustrait r2, r2, r1  
6: continue 1  
7: // fin (stop)
```

$r_1 < r_2$ → sortie si $r_1 = r_2$
 $r_1 \neq r_2$

Entrée : r_1 r_2

Sortie : r_1

```
if ( $r_1 \neq r_2$ )  
  if ( $r_1 > r_2$ )  
     $r_1 \leftarrow r_1 - r_2$   
  sortie f( $r_1, r_2$ )  
sortie  $r_1$   $r_2 \leftarrow r_2 - r_1$   
sortie f( $r_1, r_2$ )
```

► Que vaut r_1 lorsque le programme ci-dessus se termine, si au départ r_1 contenait 01000110 et r_2 00101010 ? $70 - 42 = 28$

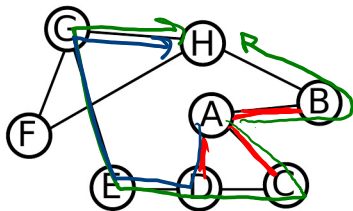
► Ecrivez l'algorithme correspondant au programme ci-dessus.

► En une phrase, que fait cet algorithme (mathématiquement) ? $42 - 28 = 14$

► Quelle est la complexité (temporelle pire cas) de cet algorithme ?

Leçon III.2 (stockage & réseaux) – Etude de cas

Laquelle des lignes suivantes fait partie de la table de routage de A pour le réseau suivant (en exprimant les distances en nombre d'arcs) :



dest	dir.	dist
H	B	2
H	C	5
H	D	4

A]

dest.	dir.	dist.
F	B	3

C]

dest.	dir.	dist.
D	G	2

B]

dest.	dir.	dist.
H	D	4

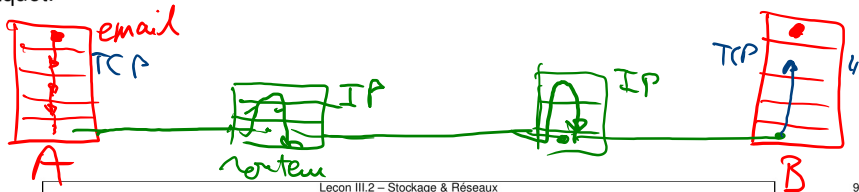
D]

dest.	dir.	dist.
A	F	3

Leçon III.2 (stockage & réseaux) – Etude de cas

Dans un réseau TCP/IP, si un paquet est perdu lors de l'échange d'emails :

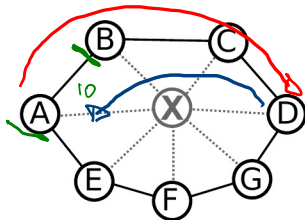
- A] Ce n'est pas grave car les emails sont redondants.
- B] Le paquet est renvoyé par l'ordinateur émetteur.
- ~~C] Le paquet est renvoyé par le dernier routeur.~~ NON
- D] C'est qu'il a été perdu par la couche TCP car la couche IP ~~garantit~~ aucune perte de paquet.



Leçon III.2 (stockage & réseaux) – Etude de cas

On considère le réseau TCP/IP suivant :

$$\frac{7}{16} \times \left(\frac{7}{16} \right)^2 \times \left(\frac{3}{4} \right)^2 \times \frac{9}{16}$$



$$\left(\frac{7}{16} \right)^n \Big|_{50=1}$$

2. On envoie un message de A vers D (X absent). Ce message est décomposé en 50 paquets. Sachant que

- ▶ chaque nœud met 10 ms à transmettre un paquet au nœud suivant ;
- ▶ on négligera tout autre temps ;
- ▶ en moyenne 1 paquet sur 4 est malheureusement perdu ;

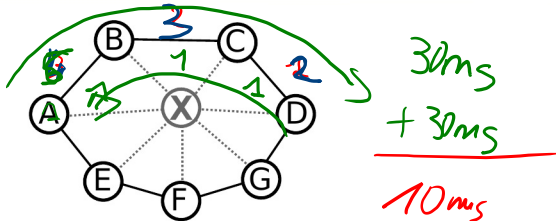
$$n=6$$

$$50 + \dots + \dots + 1$$

- quel est (en moyenne) le temps total de transmission de ce message, lorsque X est absent ?

Leçon III.2 (stockage & réseaux) – Etude de cas

On considère le réseau TCP/IP suivant :



2. On envoie un message de A vers D (X absent). Ce message est décomposé en 50 paquets. Sachant que

- ▶ chaque nœud met 10 ms à transmettre un paquet au nœud suivant ;
- ▶ on négligera tout autre temps ;
- ▶ en moyenne 1 paquet sur 4 est malheureusement perdu ;

quel est (en moyenne) le temps total de transmission de ce message, lorsque X est absent ?