

misc

Tri très sélectif ✓

104



Challenge

275 résolutions



Tri très sélectif

104

algorithmique



Comme l'épreuve **Tri sélectif**, vous devez trier le tableau, mais cette fois vous devez être efficace !

nc challenges.france-cybersecurity-challenge.fr 2052

client.py

tri-tres-sel...

Flag

Submit

Lançons (côté serveur): nc challenges.france-cybersecurity-challenge.fr 2052

```
kali@kali: ~
File Actions Edit View Help
64 bytes from 176.31.162.35: icmp_seq=32 ttl=46 time=95.5 ms
64 bytes from 176.31.162.35: icmp_seq=33 ttl=46 time=42.6 ms
64 bytes from 176.31.162.35: icmp_seq=34 ttl=46 time=44.0 ms
64 bytes from 176.31.162.35: icmp_seq=35 ttl=46 time=68.7 ms
64 bytes from 176.31.162.35: icmp_seq=36 ttl=46 time=65.9 ms
64 bytes from 176.31.162.35: icmp_seq=37 ttl=46 time=67.2 ms
64 bytes from 176.31.162.35: icmp_seq=38 ttl=46 time=45.3 ms
64 bytes from 176.31.162.35: icmp_seq=39 ttl=46 time=38.2 ms
64 bytes from 176.31.162.35: icmp_seq=40 ttl=46 time=59.4 ms
^C
 176.31.162.35 ping statistics —
 40 packets transmitted, 40 received, 0% packet loss, time 39071ms
 rtt min/avg/max/mdev = 32.950/59.015/111.115/19.311 ms

(kali@kali)~$ ping 176.31.162.35 on port 2052
ping: on: Temporary failure in name resolution

(kali@kali)~$ sudo nano /etc/hosts
[sudo] password for kali:
(kali@kali)~$ sudo nano /etc/hosts

(kali@kali)~$ ping challenges.france-cybersecurity-challenge.fr
PING challenges.france-cybersecurity-challenge.fr (176.31.162.35) 56(84) bytes of data.
64 bytes from challenges.france-cybersecurity-challenge.fr (176.31.162.35): icmp_seq=1 ttl=46 time=58.5 ms
64 bytes from challenges.france-cybersecurity-challenge.fr (176.31.162.35): icmp_seq=2 ttl=46 time=55.1 ms
64 bytes from challenges.france-cybersecurity-challenge.fr (176.31.162.35): icmp_seq=3 ttl=46 time=58.3 ms
64 bytes from challenges.france-cybersecurity-challenge.fr (176.31.162.35): icmp_seq=4 ttl=46 time=49.9 ms
64 bytes from challenges.france-cybersecurity-challenge.fr (176.31.162.35): icmp_seq=5 ttl=46 time=53.2 ms
64 bytes from challenges.france-cybersecurity-challenge.fr (176.31.162.35): icmp_seq=6 ttl=46 time=65.9 ms
^C
  challenges.france-cybersecurity-challenge.fr ping statistics —
 6 packets transmitted, 6 received, 0% packet loss, time 5008ms
 rtt min/avg/max/mdev = 49.920/56.831/65.947/5.030 ms

(kali@kali)~$ nc challenges.france-cybersecurity-challenge.fr 2052
Votre but est de trier un tableau dont vous ne voyez pas les valeurs (chacune est remplacée par *) :
*****
Actions possibles:
- "comparer X Y": compare les valeurs du tableau aux cases X et Y, et retourne 1 si la valeur en X est inférieure ou égale à celle en Y, 0 sinon.
- "echanger X Y": échange les valeurs du tableau aux cases X et Y, et affiche le tableau modifié.
- "longueur": retourne la longueur du tableau.
- "verifier": retourne le flag si le tableau est trié.
>>> ^C
```

Au début l'attention est focalisée sur la résolution DNS de nom entre domaines connectés 5G et Wifi.

Envoi d'une requête 'ping' sur challenges.france-cybersecurity-challenge.fr [176.31.162.35] avec 32 octets de données :

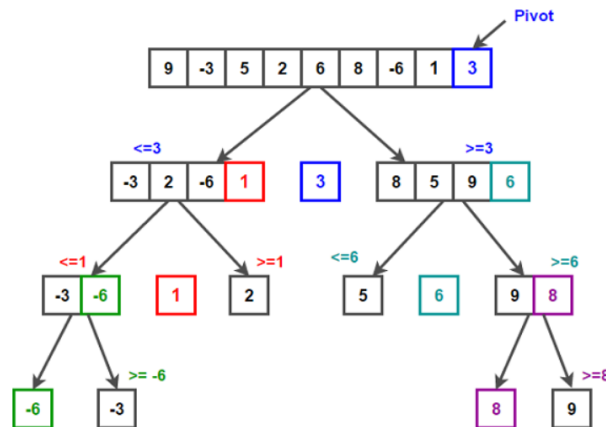
Réponse de 176.31.162.35 : octets=32 temps=17 ms TTL=47

Réponse de 176.31.162.35 : octets=32 temps=14 ms TTL=47

*

Après, l'optimisation du programme de tri fait intervenir un nouvel algorithme de tri de « quick sort ».

Quick sort partitionne un tableau à chaque étape axé sur le pivot et l'action de diviser et de conquérir.



Il s'agit d'optimisation de gestion des indices d'éléments de comparaison, en n'ayant pas les éléments.

```
kali-linux-2023.1-virtualbox-amd64 [En fonction] - Oracle VM VirtualBox
client.py - Visual Studio Code
File Edit Selection View Go Run Terminal Help
$ git.sh
client.py • tri-tres-selectif.py
home > kali > FCSC2023 > MISC > B_misc_tri_tres_selectif > client.py > quicksort
58 pivotIndex = array[0]
59 lessIndex = []
60 greaterIndex = []
61
62 index = 1
63 while(index < N):
64     if comparer(array[index], pivotIndex):
65         lessIndex.append(array[index])
66     else:
67         #print(" GREATER "+str(array[index])+" than "+str(pivotIndex))
68         greaterIndex.append(array[index])
69     index = index + 1
70 #if (count > 1):
71 return quicksort(lessIndex, 1) + [pivotIndex] + quicksort(greaterIndex, 1)
72
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
(kali@kali) - [~]
$ /bin/python /home/kali/FCSC2023/MISC/B_misc_tri_tres_selectif/client.py
[+] Opening connection to challenges.france-cybersecurity-challenge.fr on port 2052: Done
Début tri 32 valeurs
Positions apres tri
[14, 0, 23, 27, 20, 24, 19, 28, 2, 7, 6, 17, 8, 10, 12, 9, 29, 5, 15, 16, 30, 3, 21, 18, 11, 13, 31, 22, 26, 1, 25, 4]
Fin Tri
Le flag est : FCSC{6d275607ccfba86daddaa2df6115af5f5623f1f8f2dbb62606e543fc3244e33a}
[*] Closed connection to challenges.france-cybersecurity-challenge.fr port 2052
Ln 72, Col 1 (18 selected) Tab Size: 4 UTF-8 LF Python 3.11.2 64-bit CTRL DROITE
(kali@kali) - [~]
$ /bin/python /home/kali/FCSC2023/MISC/B_misc_tri_tres_selectif/client.py
[+] Opening connection to challenges.france-cybersecurity-challenge.fr on port 2052: Done
Début tri 32 valeurs
Positions apres tri
[14, 0, 23, 27, 20, 24, 19, 28, 2, 7, 6, 17, 8, 10, 12, 9, 29, 5, 15, 16, 30, 3, 21, 18, 11, 13, 31, 22, 26, 1, 25, 4]
Fin Tri
Le flag est : FCSC{6d275607ccfba86daddaa2df6115af5f5623f1f8f2dbb62606e543fc3244e33a}
[*] Closed connection to challenges.france-cybersecurity-challenge.fr port 2052
```

Le flag est affiché à l'issue de résolution du tri très rapide et de l'obtention de la liste très ordonnée :

FCSC{6d275607ccfba86daddaa2df6115af5f5623f1f8f2dbb62606e543fc3244e33a}