

Éléments de correction de l'épreuve d'admissibilité 2

Préambule : Ce sujet comporte trois parties indépendantes. Pour toutes les questions pédagogiques, vous pourrez vous appuyer sur les programmes de SNT ainsi que de première et de terminale NSI dont des extraits choisis sont donnés en Annexe.

Partie 1 - Réseaux de communication

Les notions portant sur l'Internet et les réseaux de communication font partie des programmes de l'enseignement de SNT en seconde et de NSI en première et terminale. En tant qu'enseignant ou enseignante dans ces matières, vous devrez aborder et expliquer les différentes couches de la pile protocolaire TCP/IP. Dans cet exercice, vous allez explorer certaines de ces couches.

1 Généralités sur les réseaux

1. L'Internet est basé sur une architecture en couches. Les différentes couches de cette architecture constituent la pile protocolaire utilisée par l'Internet. Expliquer pourquoi ce fonctionnement en couches a été retenu. Préciser les éventuels avantages et inconvénients d'un tel système.

L'approche en couches a été retenue afin d'avoir une grande modularité dans la conception et la mise en œuvre de l'Internet.

Une telle approche a plusieurs avantages : chaque protocole réseau est associé à une couche donnée, ce qui favorise sa conception et son identification ; la modularité simplifie l'implémentation des différentes couches (il est possible de changer l'implémentation d'une couche sans changer les autres couches).

Une telle approche a aussi des inconvénients : il peut y avoir une redondance de certaines fonctions (par exemple la détection d'erreurs sur les paquets peut se faire au niveau des couches 1, 2 et 4) ; une couche ne peut pas utiliser des informations fournies sur d'autres couches alors que ces informations pourraient lui être utiles.

2. Que définit un protocole réseau ?

Un protocole réseau permet aux éléments d'un réseau de communiquer. Un protocole réseau définit le format et l'ordre des messages échangés entre deux éléments du réseau. Il définit aussi les actions à effectuer lors de l'envoi ou la réception d'un message.

3. Expliquer ce qu'est le modèle client-serveur.

Dans une relation client/serveur, un client (qui est un élément du réseau) demande un service ou une ressource à un serveur (qui est aussi un élément du réseau).

4. Citer deux technologies de communication et donner un ordre de grandeur de leur capacité d'émission (appelé aussi débit d'émission) pour chacune d'entre elles.

Les deux technologies suivantes sont très utilisées :

- Ethernet avec une capacité d'émission entre 10 Mb/s et plusieurs dizaines de Gb/s.
- Wi-Fi avec une capacité d'émission entre une centaine de Mb/s à quelques Gb/s.

5. Vous trouverez un exemple de réseau dans le document 7. Associer à chacun des numéros légendés sur ce schéma l'un des termes suivants : Wi-Fi, Ethernet, Client, Serveur, Fibre optique, Routeur, Commutateur (aussi appelé Switch). Sur votre copie, vous pourrez ne reporter que les numéros et les termes correspondants. Il n'est pas nécessaire de reproduire le schéma.

- 1 : serveur
- 2 : commutateur
- 3 : fibre optique
- 4 : routeur
- 5 : Wi-Fi
- 6 : Ethernet
- 7 : client

6. Le programme de première de la spécialité NSI demande de présenter "le rôle des différents constituants du réseau local de l'établissement" (document 11). Inventorier les principaux éléments susceptibles de constituer un réseau local et préciser leur(s) rôle(s).

Dans un réseau d'établissement, on va trouver :

- des clients (ordinateurs ou tablettes), ils sont principalement utilisés par les personnels et les élèves ;
- des serveurs, ils permettent de mettre en place différents services (authentification, partage de ressources (fichiers, imprimante, ...)) ;
- des commutateurs (appelés aussi switches), ils permettent de relier plusieurs ordinateurs afin de constituer un réseau local ;
- un routeur (passerelle) qui permet de relier le réseau de l'établissement au réseau extérieur (Internet).

2 Couche transport

7. Quel est l'objectif d'un protocole de transport ?

L'objectif d'un protocole de transport est de gérer la communication de bout-en-bout.

8. Expliquer le principe général de fonctionnement du protocole de transport TCP.

Le protocole de transport TCP nécessite l'établissement d'une connexion entre les deux entités communiquant avec ce protocole. Une fois la connexion établie, TCP assure la fiabilité de la communication. Pour cela le destinataire doit acquitter auprès de l'émetteur la bonne réception des segments TCP. Si l'émetteur ne reçoit pas, au bout d'un certain temps, d'acquittements sur certains segments TCP envoyés, il va les retransmettre.

9. Comparer le protocole de transport UDP et le protocole de transport TCP. Donner les avantages et les inconvénients de chacun de ces protocoles.

UDP n'établit pas de connexion et ne garantit pas la fiabilité de la communication, à l'inverse de TCP. UDP est rapide alors que TCP est plus lent. En revanche, TCP est fiable alors que certains paquets peuvent ne jamais arriver avec UDP.

10. Donner un exemple d'application reposant sur le protocole de transport UDP et un exemple d'application reposant sur le protocole de transport TCP.

Les applications Web basées sur le protocole HTTP reposent sur TCP. Les applications de Téléphonie sur IP reposent sur UDP.

11. Le protocole de transport TCP est au programme de SNT (document 8). Proposer une activité débranchée permettant de faire comprendre aux élèves les notions de fiabilité d'une transmission et d'absence de garantie temporelle caractérisant le protocole TCP. L'activité proposée devra mettre en évidence les problèmes induits par cette absence de garantie temporelle.

Voici un exemple d'activité parmi d'autres possibilités : deux élèves s'échangent des messages à l'aide de petits morceaux de papier. Puisque l'on cherche à illustrer la notion de fiabilité des transmissions, on mettra en place un système d'accusé de réception. On simulera les pertes de messages grâce à un troisième élève qui "prélèvera" de manière plus ou moins aléatoire le message ou l'accusé de réception. Les trois élèves devront essayer mettre en œuvre toutes les situations possibles. À la fin de l'expérience, les élèves devront produire un compte-rendu. On attend que les élèves arrivent à la conclusion suivante : grâce à TCP un paquet finira toujours par arriver à destination (sauf cas de panne majeure), en revanche TCP ne peut pas garantir le temps mis par un paquet pour arriver à destination (comme ils auront pu l'observer lors des différentes situations identifiées). Il est possible d'amener les élèves à ces conclusions à l'aide d'un questionnaire de l'enseignement à la fin de l'activité décrite ci-dessus.

3 Couche réseau

12. Quel est l'objectif d'un protocole de la couche réseau ?

L'objectif principal d'un protocole de la couche réseau est de permettre le routage des paquets dans un réseau afin que les paquets atteignent leur destinataire.

3.1 Adressage IP

13. Décrire une activité que vous proposez à vos élèves de seconde en SNT pour expliquer le but d'une adresse IP et comment elle est organisée.

Il est possible tout d'abord d'introduire la notion d'adresse IP, par exemple en faisant l'analogie avec une adresse postale (adresse qui dépend de la localisation de la machine dans le réseau, adresse unique et hiérarchique). Pour familiariser les élèves avec la notion d'adresse IP il est possible de montrer des adresses IP en surveillant les paquets entrant ou sortant d'une carte (interface) réseau. On peut alors analyser l'organisation d'une adresses IP en expliquant la notion de partie réseau et de partie hôte dans une adresse IP. On pourra aussi mentionner la notion de masque sur des cas simples (par exemple /8, /16 ou /24). Enfin il est possible de faire manipuler aux élèves les adresses IP grâce aux commandes ifconfig et ping.

14. Expliquer le rôle du protocole DHCP.

Le protocole DHCP (Dynamic Host Configuration Protocol) permet d'assurer la configuration automatique des paramètres IP d'une machine, notamment en lui attribuant automatiquement une adresse IP et un masque de sous-réseau.

3.2 Routage

Protocoles de routage

15. Le routage est au programme de la spécialité NSI terminale (document 13). Proposer un plan de cours sur ce sujet en 10 lignes maximum, puis décrire une activité que vous réaliserez avec vos élèves.

Voici un plan de cours possible :

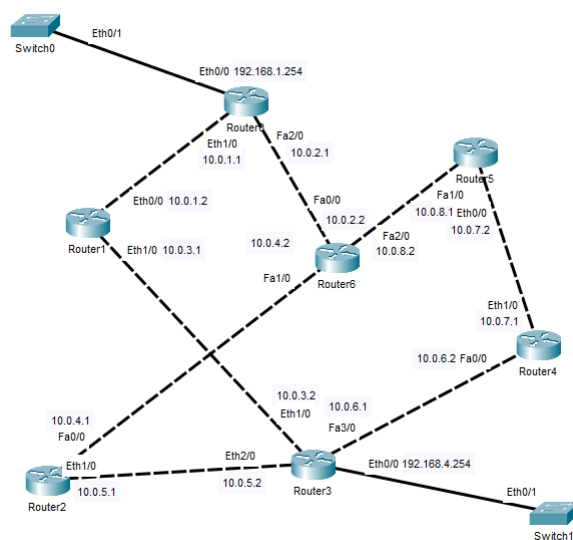
- Donner les objectifs généraux du routage. Rappeler que ça concerne la couche Réseau de l'Internet.
- Montrer que dans un réseau, il peut exister plusieurs chemins entre une machine émettrice et une machine destinatrice. Faire comprendre que ces chemins n'offrent pas forcément les mêmes performances et expliquer à quoi peut correspondre la notion de métrique sur chaque lien composant un chemin.
- Expliquer que l'objectif du routage est donc de trouver un plus court chemin entre toute paire nœuds dans le réseau.
- Expliquer l'approche du routage par état de liens. Dérouler l'approche sur un exemple. Indiquer qu'OSPF est un protocole de routage par état de liens.
- Expliquer l'approche du routage par vecteurs de distance. Dérouler l'approche sur un exemple. Indiquer que RIP est un protocole de routage par vecteurs de distance.
- Comparer les deux approches.

En termes d'activité, on peut faire travailler les élèves plus spécifiquement sur une approche de routage. Par exemple, on peut les faire travailler sur l'approche par vecteurs de distance et sur RIP. On pourra, à partir d'un schéma réseau (machines hôtes reliés par des routeurs), demander aux élèves d'établir une table de routage simplifiée (une colonne "ce que je cherche à atteindre", une colonne "moyen de l'atteindre" et une colonne métrique). On demande ensuite aux élèves de compléter la colonne "métrique". On pourra les guider en indiquant un ordre sur les messages échangés. On pourra ainsi montrer que l'ordre des messages a un impact sur le temps de convergence des tables de routage.

16. Expliquer le principe du protocole OSPF. Quel algorithme de parcours est utilisé par ce protocole? Illustrer le fonctionnement de cet algorithme sur un schéma réseau que vous choisirez.

Un routeur utilisant le protocole OSPF collecte l'ensemble des coûts des liens (en envoyant à intervalle régulier des messages "Hello" à ses voisins immédiats). Ces messages Hello sont diffusés dans tout le réseau grâce à un mécanisme d'inondation. À partir de sa connaissance de la topologie du réseau, le routeur détermine, grâce à l'algorithme de Dijkstra, un plus court chemin vers chaque nœud possible de la topologie. Ces meilleures routes sont alors intégrées à la table de routage qui comprend, pour chaque nœud destinataire de la topologie, le nœud suivant à qui envoyer les paquets ainsi que le coût du chemin pour atteindre ce destinataire.

Pour illustrer le fonctionnement de cet algorithme, il était attendu que le ou la candidate propose un schéma avec plusieurs routeurs et que l'algorithme de Dijkstra soit déroulé sur cet exemple. La figure 1 donne un exemple de réseau (les adresses IP des différents ports des routeurs n'étaient pas attendues). La figure 2 donne un déroulé de l'algorithme de Dijkstra sur le réseau de la figure 1. La matrice d'adjacence indique quels sont les routeurs voisins et les coûts des liens. L'algorithme de Dijkstra est appliqué sur le routeur R0 (qui est donc considéré comme sélectionné). À la première étape, l'algorithme sélectionne, parmi les routeurs voisins de R0, le routeur qui a le plus petit coût. C'est le routeur R6. À la deuxième étape, l'algorithme examine les routeurs voisins de R6 qui n'ont pas déjà été sélectionnés par l'algorithme et choisit celui qui a le plus petit coût. Comme les deux routeurs voisins de R6 ont le même coût, ils peuvent être sélectionnés tous les deux. Si la convention "en cas de conflit, on choisit le routeur avec le plus petit label", alors c'est le routeur R2 qui est sélectionné. Le déroulé complet de l'algorithme est indiqué dans la figure 2.



DOCUMENT 1 – Exemple de réseau.

La recherche d'un plus court chemin sur un graphe est une problématique sous-jacente à celle du routage de données. Dans la section suivante, on s'intéresse à l'enseignement de notions relatives aux graphes (programme de la classe de STI, document 9) et à leur parcours (programme de la classe de terminale NSI, document 16).

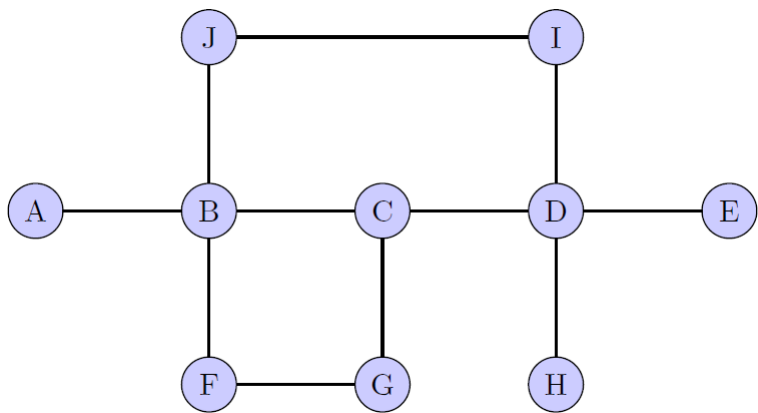
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	
1	Matrice d'adjacence									Sommet pivot	--	Sommets												Etape n°					
2	d\	R0	R1	R2	R3	R4	R5	R6		Valeur pivot	0		R0	R1	R2	R3	R4	R5	R6										
3	R0		10					1		Sommet départ	R0		--	0	R0	10										R0	1	1	
4	R1	10			10					Num départ	1															R6	2	R0	1
5	R2				10			1							R6	2	R2	12											
6	R3		10	10		1														R5	12			R6	2				
7	R4				1		10							R0	10				R1	20									
8	R5					10		1										R2	12	R3	13								
9	R6	1		1			1													R5	12								
10																													
11													--	0	R0	10									R0	1			

DOCUMENT 2 – Déroulé de l'algorithme de Dijkstra sur le réseau de la figure 1.

Utilisation des graphes

17. Proposer un exercice destiné à des élèves de seconde permettant de travailler les notions de rayon, diamètre et centre d'un graphe. Vous rédigerez une correction de cet exercice.

Voici un exemple d'exercice : un réseau social de dix individus est modélisé par le graphe donné en figure 3. Chaque individu est représenté par l'un des sommets de A à J. Les liens d'amitié sur ce réseau social sont représentés par les arêtes de ce graphe. On appelle distance entre deux individus du graphe le nombre minimum de liens d'amitié adjacents à parcourir pour relier ces deux individus.



DOCUMENT 3 – Exemple de réseau social modélisé par un graphe.

- Q1. Pour chaque individu du réseau social, déterminer la distance le reliant à chaque autre individu. Présenter les résultats dans un tableau à double entrée.
- Q2. Donner deux individus parmi les plus éloignés dans ce réseau social.
- Q3. Selon vous, quel individu de ce réseau social pourrait être le plus influent ? Argumenter.

On appelle diamètre la plus grande distance relevée parmi toutes les paires de sommets du graphe. On appelle rayon du graphe la plus distance possible entre un sommet de référence et tous les autres sommets du graphe. Tout sommet de référence pouvant être retenu pour déterminer le rayon du graphe est appelé centre du graphe.

- Q4. Relier les notions de diamètre, rayon et centre aux questions traitées précédemment.
- Q5. Insérer un nouvel individu sur le graphe de façon à avoir deux centres bien distincts.

Voici un corrigé de cet exercice :

- Q1. Les distances entre les individus sont donnés en figure 4.
- Q2. La distance la plus importante relevée dans le tableau précédent est 4. On l'observe sur le graphe entre les individus A et E, ou encore entre les individus F et H.
- Q3. On pourrait penser à l'individu C. En effet, s'il partage une opinion sur le réseau social, il suffit qu'elle soit relayée au plus par chacun de ses amis pour que chacun individu du réseau en soit informé.
- Q4. Le diamètre est la notion traitée à la question 2. Il vaut 4. Les notions de rayon et centre sont implicitement évoquées à la question 3. En effet, l'individu le plus influent est celui qui est relié à l'ensemble des autres membres du réseau en une distance minimale. Il s'agit donc d'un centre du graphe. La distance minimale nécessaire pour le relier à chaque autre individu est le rayon, de valeur 2.
- Q5. On peut par exemple ajouter un individu K, ami avec B, C, D et G.

A	B	C	D	E	F	G	H	I	J	K
A										
B										
C										
D										
E										
F										
G										
H										
I										
J										

DOCUMENT 4 – Distances entre les individus du réseau social donné en figure 3.

18. Le document 15 est un exercice proposé à des élèves de terminale en spécialité NSI.

- (a) Proposer une correction et un barème de cet exercice en justifiant votre barème.

Correction de l'exercice III (document 9)

- i. On donne ici une implémentation à l'aide d'un dictionnaire et de listes :

```
graph={ 'A': [ 'B', 'F' ], 'B': [ 'A', 'D', 'E' ], 'C': [ 'E', 'G' ],
        'D': [ 'B', 'G' ], 'E': [ 'B', 'C', 'F' ], 'F': [ 'A', 'E' ], 'G': [ 'C', 'D' ] }
```

- ii. Voici un exemple de parcours en profondeur d'abord à partir du sommet A : A-F-E-C-G-D-B
 iii. Voici un exemple de parcours en largeur d'abord à partir du sommet A : A-B-F-D-E-G-C
 iv. Voici un algorithme permettant de réaliser un parcours en largeur d'abord :

VARIABLES

G : un graphe
 s : noeud (origine)
 u : noeud
 v : noeud
 f : **file** (initialement vide)

//On suppose que tout sommet u du graphe G a initialement la couleur blanc, u.couleur

DEBUT

```
s.couleur ← noir
enfiler (s,f)
tant que f non vide :
  u ← defiler(f)
  pour chaque sommet v voisin du sommet u :
    si v.couleur n'est pas noir :
      v.couleur ← noir
      enfiler(v,f)
  fin si
fin pour
fin tant que
FIN
```

On peut également accepter une implémentation en langage Python. Ci-dessous une classe "file" est créée pour enfiler les sommets successivement parcourus. Les sommets colorés (déjà parcourus) sont conservés dans un objet de type set.

```
class file(list):
    curseur = -1
    color = set()

    def est_vide(self):
        return self.curseur == len(self) - 1

    def enfile(self, elt):
        self.append(elt)
        self.color.add(elt)
```

```

def defile(self):
    self.curseur += 1
    return self[self.curseur]

def parcours_largeur(g):
    f = file()
    f.enfile(next(iter(g)))
    while not f.est_vide():
        u = f.defile()
        for v in g[u]:
            if not v in f.color:
                f.enfile(v)
    return f

```

Voici un exemple de barème : Q1 - deux points peuvent être attribués de la manière suivante : 1 pt pour la pertinence de la structure retenue pour l'implémentation et 1 pt pour son exploitation dans le cadre de l'exercice donné. ; pour Q2 et Q3, un point par question, selon que le principe de parcours en largeur ou profondeur est compris ou non ; quatre points peuvent être attribués à la question Q4 de la manière suivante : 1 pt si l'algorithme enfile les sommets par ordre croissant de profondeur - 1 pt si l'algorithme n'enfile pas plusieurs fois le même sommet - 1 pt si l'algorithme parcourt correctement les structures de données manipulées (intégralement et sans dépassement) et 1 pt si les entrées et sorties de l'algorithme sont correctes.

(b) Relever les erreurs éventuelles de la copie donnée dans le document 17.

- Q1 : la structuration en dictionnaire et listes proposée par l'élève est correcte, mais plusieurs éléments manquent comme les clés F et G, ou encore le sommet A parmi la liste des sommets associés à la clé B. Ces manques sont susceptibles de traduire un problème méthodologique dans le parcours de l'ensemble des arêtes du graphe.
- La question Q2 est correctement traitée.
- Dans la question Q3, l'élève parcourt le sommet E avant le sommet F. Soit le principe du parcours en profondeur est mal connu, soit sa réalisation technique (via file ou marquage) est incorrecte.
- Q4 : l'élève utilise une pile à la place d'une file, ce qui ne permet pas de faire un parcours en largeur.

(c) Proposer des améliorations à cet exercice afin, notamment, de le rendre plus accessible pour un élève ayant un niveau "moyen" en spécialité NSI de terminale.

Ce type d'exercice se révèle difficile puisqu'il demande à l'élève de concevoir une structure modélisant les données de l'exercice et qui soit adaptée à la résolution algorithmique du problème de parcours en largeur. Plusieurs pistes peuvent être retenues pour faciliter l'accès à cet exercice : proposer, en début d'exercice, de mettre en application une technique de marquage pour chacun des parcours ; à cette occasion l'exercice peut également proposer de remplir à la main un tableau (ou une file) pour illustrer la structuration machine des données ; l'enseignant peut proposer une discussion entre élèves sur la façon de structurer les données (en Python ou en langage naturel) pour que la traduction algorithmique du parcours en largeur soit la plus cohérente possible.

4 Étude du fonctionnement général et des performances d'un réseau

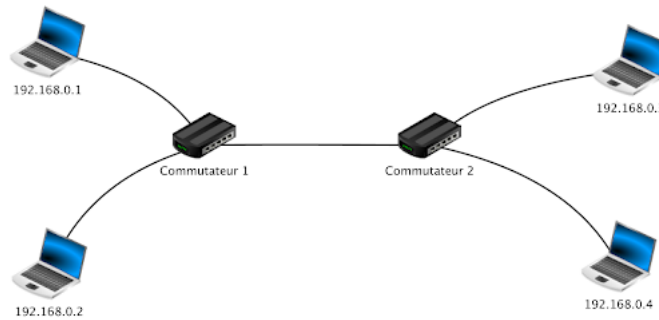
Cette partie concerne le programme de première de la spécialité NSI, donné dans le document 11.

19. Dans ce programme, il est demandé de "simuler ou mettre en œuvre un réseau". Proposer une séquence pédagogique où les élèves auront à mettre en œuvre un réseau à l'aide d'un logiciel de simulation. Vous préciserez le logiciel que vous comptez utiliser. Proposer un schéma du réseau qui serait simulé et décrire les activités que vous proposeriez sur ce scénario réseau.

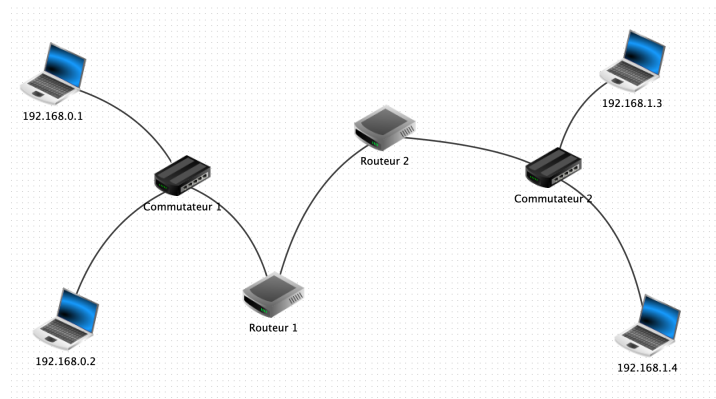
Le logiciel Filius permet de découvrir un ensemble de notions fondamentales relatives aux réseaux et au routage des données. Pour des simulations plus évoluées, un simulateur professionnel comme Cisco Packet Tracer peut être envisagé.

Plusieurs réseaux, de complexité croissante peuvent être étudiés par les élèves. Tout d'abord on peut envisager de leur faire connecter deux machines. À cette occasion les concepts d'IP et de masque de sous-réseau peuvent être introduits. On peut ensuite travailler sur un réseau local avec une architecture plus complexe comprenant des machines reliées par un ou plusieurs commutateurs (switch) dont un exemple est donné en figure 5. À cette occasion, on peut introduire la problématique de la distribution des adresses IP et présenter le rôle d'un serveur DHCP.

On peut enfin finir avec une architecture reliant plusieurs réseaux locaux par un ou plusieurs routeurs, dont un exemple est donné en figure 6. À cette occasion, les concepts de passerelle et de route peuvent être introduits.



DOCUMENT 5 – Simulation d'un réseau local.



DOCUMENT 6 – Simulation d'un réseau avec routeurs.

Au travers des différents réseaux proposés, il est attendu des élèves qu'ils apprennent à configurer les adresses IP des interfaces réseaux des équipements, et à tester leur configuration. En cas d'échec, développer l'aspect méthodologique visant à analyser les défaillances d'un réseau est recherché. Les commandes comme Ping ou Tracert peuvent être exploitées judicieusement à cet effet.

20. Discuter les avantages et les inconvénients de la simulation pour étudier le fonctionnement ou les performances d'un réseau.

La simulation permet de travailler sur des réseaux variés sans avoir besoin de matériel réel. De plus, pour des raisons de sécurité, il est difficile de faire administrer par les élèves un réseau de l'établissement scolaire (à moins d'avoir suffisamment de matériel pour s'équiper d'un réseau en propre isolé des réseaux de l'établissement), et la simulation permet donc de contourner ce problème. En revanche, la simulation n'est toujours qu'un modèle de la réalité et tous les éléments d'un vrai réseau (certains protocoles, certaines technologies de communication, certains matériels) ne sont pas intégrés dans le simulateur.

21. Le programme demande de "mettre en évidence l'intérêt du découpage des données en paquets et de leur encapsulation". Citer les facteurs qui ont un impact sur le temps de transfert d'un message de la source à la destination. À partir de ces facteurs, expliquer l'intérêt d'un découpage des données à transmettre en paquets. À quoi peut conduire un découpage excessif (à savoir en un très grand nombre de paquets) ?

Voici des exemples de facteurs qui ont un impact sur le temps de transfert d'un message de la source à la destination :

- le nombre de liens traversés par le message sur le chemin reliant la source à la destination ;
- la capacité d'émission et le délai de propagation de chaque lien traversé ;
- la fiabilité de chaque lien traversé (est-ce que des paquets sont perdus lors de la transmission à cause d'erreurs de transmission ?) ;
- la taille du message à transmettre ;
- le surcoût introduit dans le message lié à l'encapsulation du message dans les différentes couches protocolaires.

En découpant le message à envoyer en plusieurs plus petits paquets, les liens (constituant le chemin de la source à la destination) peuvent alors émettre en parallèle (par ex. le lien k peut envoyer le paquet i pendant que le lien $k + 1$ transmet le paquet $i - 1$, etc.). Une telle parallélisation permet de réduire le temps de transfert du message comparé à la transmission intégrale du message qui, elle, ne peut se faire que lien après lien de manière séquentielle. D'autre part, si un paquet est perdu, il suffit seulement de retransmettre le paquet perdu (si on est dans le cadre d'un service réseau fiable) et non le message dans son intégralité.

Un découpage excessif des paquets (en de trop nombreux petits paquets) peut entraîner un surcroît de travail au niveau de la machine émettrice (découpage des données) et de la machine réceptrice (reconstitution des données), ainsi qu'un surcoût au niveau des en-têtes (lié à l'encapsulation).

Partie 2 - Le Web

Cette partie porte sur le World Wide Web qui sera abrégé en Web dans la suite de l'énoncé.

5 Généralités, langages et URL

Le Web fait partie du programme de SNT comme indiqué dans le document 10.

22. Donner une définition du Web.

Le Web est un ensemble de contenus accessibles via le réseau Internet. On accède, via un navigateur, à un ensemble de ressources Web identifiées par des URL et mises en lien par des hyperliens.

23. À quelle date et par qui le Web a-t-il été "inventé" ?

Le World Wide Web a été inventé dans les années 1990 par Tim Berners-Lee et Robert Cailliau au CERN.

24. Proposer une séquence pédagogique permettant d'introduire le HTML et le CSS en classe de SNT.

La séquence pédagogique pourrait s'articuler en trois parties : i) les élèves peuvent démarrer avec une utilisation d'"outils de développement" des navigateurs web afin d'étudier le code source d'une page web "simple" (HTML et CSS). L'enseignant ou l'enseignante peut ensuite proposer quelques modifications du code source, toujours depuis l'outil de développement du navigateur, afin que les élèves commencent à s'appropriier les mécanismes mis en jeu ; ii) à la suite de cette activité pédagogique, les éléments de base du HTML et du CSS ont alors enseignés ; iii) pour terminer cette séquence, on demande aux élèves de réfléchir à des modifications de la page étudiée initialement et de faire leurs propres modifications.

25. À la suite de votre enseignement sur HTML et CSS, vous comptez évaluer vos élèves via un QCM. Proposer un QCM comportant 3 questions ayant chacune 4 choix possibles et un seul choix correct. On justifiera le choix des questions et des réponses proposées, en précisant la réponse juste.

Les questions proposées doivent pouvoir permettre d'identifier si les élèves ont bien compris quels sont les usages relatifs à chacun des deux langages. Les questions suivantes peuvent être proposées :

Q1 : Le langage HTML permet de

- (a) gérer des masses de données
- (b) créer un serveur web
- (c) structurer une page web et son contenu
- (d) envoyer des informations sur Internet

Q2 : Parmi les codes suivants, lequel est à préférer ?

- (a) Un fichier index.html contenant :

```
<head>
<style>p text-align :center ;</style>
</head>
<body>
<p>NSI</p>
</body>
```

- (b) Un fichier index.html contenant :

```
<head>
<link href="style.css" rel="stylesheet" type="text/css">
</head>
<body>
<p>NSI</p>
</body>
```

Et un fichier style.css contenant :

```
p { text-align :center ; }
```

- (c) Un fichier index.html contenant :

```
<body>
<p style="text-align :center">NSI</p>
</body>
```


- (d) Un fichier index.html contenant :
- ```
<body>
<center><p>NSI</p></center>
</body>
```

Q3 : Une feuille de style CSS contient le code suivant : `.infos { color : red; }` Elle est importée dans un fichier HTML via la balise `link`. Quel code, placé dans ce fichier HTML affichera NSI en rouge ?

- (a) `<div class="infos" class="element">NSI</div>`  
(b) `<div class="infos">NSI</div><div class="element">NSI</div>`  
(c) `<div class="element"><div class="infos">NSI</div></div>`  
(d) `<div class="infos"><div class="element">NSI</div></div>`

Discussion sur les différentes réponses : la réponse juste de la question Q1 est la (c). L'objectif de cette question est de vérifier que les élèves connaissent le but du langage HTML et qu'ils ne confondent pas avec l'applicatif fournissant des pages web (serveur web), ni avec le protocole transportant des pages web (protocole HTTP); la réponse juste de la questions Q2 est la (b). L'objectif de cette question est de vérifier que les élèves ont bien compris la nécessité de dissocier l'ordonnancement du texte (réservé au HTML) et l'établissement des caractéristiques de style (réservé au langage CSS), les différentes réponses proposées réalisant plus ou moins cette séparation; la réponse juste de la questions Q3 est la (d). Cette question demande aux élèves de savoir analyser quelles sont les éventuelles classes ou sous-classes associées à chacun des items renseignés sur le document HTML. Il permet de vérifier que la correspondance entre la structuration des classes exploitées dans un document HTML et dans la feuille de style associée est correctement comprise.

26. Expliquer la structure générale d'une URL.

Pour illustrer la structure générale d'une URL, on retient l'exemple suivant : `http://www.monSite.fr/nvx/accueil.html` :

- "`http`" : il s'agit du protocole envoyé
- "`www.monSite.fr`" : adresse symbolique du site appelée aussi nom de domaine
- "`/nvx/accueil.html`" : chemin vers le fichier demandé sur le site contenant le contenu

## 6 HTTP et HTTPS

27. À quoi sert le protocole HTTP ?

L'Hypertext Transfer Protocol est un protocole applicatif de type client-serveur permettant un client web (navigateur) de dialoguer avec un serveur web. Le protocole en lui-même est conçu de façon à pouvoir renseigner le client sur le contenu de ressources web (texte enrichi, contenus multimédia), mais aussi sur l'existence de ces ressources ou encore sur leur droit d'accès.

28. Expliquer le rôle des méthodes GET et POST du protocole HTTP.

La méthode GET permet de demander une ressource Web et donc de solliciter la transmission de la ressource demandée. La méthode POST est utilisée pour transmettre des données à un serveur en vue de créer ou de mettre à jour une ressource web.

29. Proposer une séquence pédagogique permettant d'aborder le protocole HTTP avec des élèves de la spécialité NSI de première (document 12).

Il peut être intéressant de mettre à disposition un site web élémentaire, contenant un formulaire, sur le réseau local de l'établissement scolaire.

Chaque élève pourra utiliser les outils de développement web d'un navigateur pour observer le contenu des entêtes des requêtes HTTP effectuées vers le site proposé, ainsi que des des réponses retournées. Différentes manipulations peuvent être proposées pour mettre en évidence des codes de réponse HTTP, ainsi que leur rôle, comme par exemple les codes 200 (contenu disponible), 301 (contenu déplacé), 403 (accès interdit), ou 404 (contenu absent). De même l'utilisation des requêtes GET et POST pourra être exploitée afin de distinguer les usages en terme de dialogue sur le web. La distinction entre transmission de contenu et récupération de contenu sera reliée à l'utilisation de ces deux requêtes.

30. La sécurisation des communications est au programme de terminale NSI (document 13). Expliquer les principes :

- du chiffrement symétrique,

Dans le cas d'un chiffrement symétrique, 2 interlocuteurs, A et B vont chiffrer leurs messages en utilisant une clé de chiffrement commune, avant de les transmettre à l'autre. Ils ont également en leur possession une clé de déchiffrement commune pour pouvoir lire les messages reçus. Les clés doivent être connues uniquement de A et B, ce qui pose le problème de leur échange avant de pouvoir communiquer.

- du chiffrement asymétrique,

Dans le cas d'un chiffrement asymétrique, A possède une clé publique `pukA` et une clé privée `prkA`. Il en est de même pour B qui possède une clé publique `pukB` et une clé privée `prkB`. Quand A désire envoyer un message

m à B, il chiffre ce message m avec la clé  $\text{pk}_B$  (qui est accessible à tout le monde), il obtient alors un message chiffré à transmettre à B, qui lui le déchiffre avec sa clé  $\text{pr}_B$  (connue uniquement de lui). Le principe est identique lorsque B souhaite envoyer un message à A.

- du protocole HTTPS (on s'intéressera uniquement à la partie "sécurité" de ce protocole).

Le protocole HTTPS permet de chiffrer les communications entre un client et un serveur. Voici les étapes de ce processus :

- Le client effectue une requête HTTPS vers le serveur, en retour le serveur envoie sa clé publique ( $\text{K}_{\text{puS}}$ ) au client.
- Le client génère une clé K (qui sera utilisée pour chiffrer les futurs échanges), chiffre cette clé K avec  $\text{K}_{\text{puS}}$  et envoie la version chiffrée de la clé K au serveur.
- Le serveur reçoit la version chiffrée de la clé K et la déchiffre en utilisant sa clé privée ( $\text{K}_{\text{prS}}$ ). À partir de ce moment-là, le client et le serveur sont en possession de la clé K.
- Le client et le serveur commencent à échanger des données en les chiffrant et en les déchiffrant à l'aide de la clé K (chiffrement symétrique).

31. Vous désirez mettre en place une activité permettant d'illustrer le principe de chiffrement symétrique auprès de vos élèves de spécialité terminale NSI. Décrire l'activité que vous allez réaliser et motiver les choix que vous avez faits.

Il est possible de proposer aux élèves une activité débranchée sur le chiffrement symétrique. Voici un exemple :

- Les élèves travaillent en trinôme (A, B et C). Deux des trois élèves (A et B) se mettent d'accord sur une clé de chiffrement K (cette clé est un mot qui sera "traduit" en code ASCII (binaire)). C n'a pas connaissance de cette clé.
- A choisit un mot M. il "traduit" les caractères composant M en code ASCII (binaire).
- A chiffre M en appliquant un XOR entre M et K. Il obtient M' (M' est "traduit" en une succession de caractères à partir du code binaire).
- A donne M' à B et à C.
- B et C essaient de retrouver le message M (B avec la clé K et C sans cette clé).

Cette activité comporte plusieurs intérêts. Elle permet de réinvestir les notions abordées en 1ère NSI en logique booléenne et sur la représentation des données. Ensuite, elle permet de mettre en pratique un chiffrement d'apparence robuste. Des échanges avec les élèves peuvent être initiés afin de les amener à s'interroger sur cette apparente robustesse dans les cas suivants : i) K est un mot d'une lettre; ii) K est un mot de même taille que M. Les problématiques induites par ces cas extrêmes (attaque statistique ou problématique de la transmission de la clé de chiffrement) peuvent être évoquées afin de justifier les techniques cryptographiques modernes.

## 7 Moteur de recherche

L'étude des moteurs de recherche est au programme de SNT (document 10).

32. À quoi sert un moteur de recherche ?

Un moteur de recherche est une application web permettant de retrouver des informations stockées sur le Web. L'utilisateur spécifie des mots clés permettant de guider la recherche.

33. Expliquer le principe général de fonctionnement de l'algorithme "PageRank" qui est à la base du moteur de recherche de Google.

PageRank est un algorithme utilisé par Google qui permet de mesurer l'importance d'un site ou d'une page web. Pour une page web donnée, cet algorithme détermine une note en effectuant un calcul basé sur le nombre de liens pointant vers cette page et sur l'importance relative des autres pages la référençant. Le PageRank d'une page web est donc définie récursivement et dépend des métriques PageRank des pages web pointant sur elle. D'autres éléments sont également retenus dans le calcul des notes pour affiner leur valeur (analyse du surf des utilisateurs, analyse des contenus, l'utilisation d'HTTPS), mais le fonctionnement complet de PageRank n'est connu que de Google.

34. Vous demandez à trois élèves de seconde de réaliser un exposé sur les moteurs recherches. Proposer une grille d'évaluation pour cet exposé.

La grille d'évaluation proposée doit prendre en compte à la fois les aspects techniques (notamment sur le fonctionnement interne du moteur de recherche, quand celui-ci est connu ; exemples : principe de PageRank, indexation...) et sociétaux (modèle économique, respect de la vie privée...). On pourrait évaluer les éléments suivants :

- la description technique des moteurs présentés et la compréhension de ces aspects techniques ;
- la prise en compte d'aspects sociétaux ;
- la prise de recul concernant l'usage de ces outils ;
- la qualité de la présentation orale et des supports éventuels de présentation ;

## Partie 3 - Développement d'applications

Une partie de l'enseignement de NSI est consacrée à l'élaboration de projets conduits par les élèves, comme cela est précisé dans le document 14. On s'intéresse, dans la suite, à deux projets proposés par deux groupes d'élèves de terminale NSI.

### 8 Logiciel d'emprunt de livres

Un groupe d'élève souhaite réaliser un logiciel d'emprunt de livres. Le cahier des charges établi est le suivant :

- Les informations relatives aux utilisateurs et aux livres disponibles à l'emprunt doivent être structurées dans une base de données.
- Le modèle conceptuel de données retenu est celui fourni dans le document 19.
- Le logiciel en lui-même doit être muni d'une interface graphique.

35. Justifier le choix d'une structure de base de données dans le cadre de ce projet.

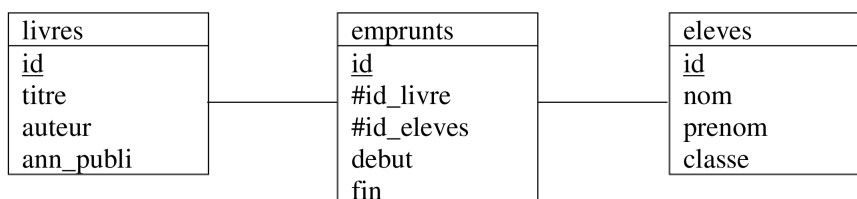
Il se justifie par la rapidité d'accès aux données, la gestion de la cohérence des données ou encore la persistance des transactions.

36. Proposer un langage permettant d'implémenter une interface graphique. Recommanderiez-vous l'usage d'une bibliothèque de ce langage à cette fin ? Si oui, laquelle ?

Il est possible d'utiliser le langage Python. L'usage des bibliothèques Tkinter, Gtk, Qt ou wx peut être recommandé.

37. Un élève du groupe choisit de se consacrer à la structuration de la base de donnée et implémente les trois tables fournies dans le document 20. De quoi cet élève n'a-t-il pas tenu compte ? Corriger sa proposition.

L'association non hiérarchique présente sur le modèle conceptuel de données (Document 13) n'a pas été traduite sur le modèle logique de données (Document 14). Dans la table "emprunts", il devait apparaître deux clés étrangères : "id\_livre" (correspond à la clé "id" de la table "livres") et "id\_eleve" (correspond à la clé "id" de la table "eleves"). Selon le SGBDR utilisé, l'ajout d'une clé primaire sous la forme d'un champ unique à la table "emprunts" peut s'avérer nécessaire. Une représentation correcte pourrait être :



38. Présenter le concept de clé étrangère tel que vous le feriez à des élèves de terminale NSI.

Une clé étrangère est un champ (ou un ensemble de champs) d'une table dont le contenu des enregistrements fait référence aux enregistrements d'un champ (ou d'un ensemble de champs) d'une clé primaire d'une autre table. Elle permet de mettre en lien des données de deux tables et de s'assurer de leur intégrité.

Sur l'exemple suivant, pouvant être fourni aux élèves, on met en évidence l'intérêt de scinder une table en deux pour éviter une redondance des données.

Supposons que la table "livres" comporte en plus un champ "date\_naissance\_auteur", on a une redondance des données puisque l'on peut trouver plusieurs fois le même auteur dans les enregistrements de la table. On propose alors aux élèves de séparer la table "livres" en 2 tables : "livres" et "auteurs" et de créer un lien entre les deux tables : la clé étrangère.

39. Proposer trois situations que les élèves de ce groupe pourront rencontrer lors de la réalisation de leur projet qui amènent à réaliser :

(a) Une requête de sélection de données sur la table "livres".

Un utilisateur désire connaître l'auteur d'un livre : `SELECT auteur FROM livres WHERE titre='BladeRunner'`

(b) Une requête d'insertion de données sur la table "livres".

On désire ajouter un livre à la collection : `INSERT INTO livres VALUES (356,'1984','Orwell',1949)`

(c) Une requête de mise à jour de données portant sur une jointure des tables "livres" et "emprunts".

Un utilisateur désire prolonger un emprunt : `UPDATE emprunts INNER JOIN livres ON emprunts.id_livre = livres.id SET emprunts.fin = '2021-03-23' WHERE livres.nom='1984'`

Fournir également ces requêtes en langage SQL.

40. Proposer une grille d'évaluation pour ce projet.

Le tableau suivant recense différents items évaluable dans le cadre de ce projet.

Compétence	Items évalués
Langage Python	Maitrise du code Gestion des événements
Bases de données	Pertinence de la structuration Robustesse des transactions
Génie logiciel	Adéquation entre fonctionnalités réalisées et attendues d'un tel logiciel
Production collaborative	Découpage du code Utilisation d'outils de développement collaboratif (git, svn) Documentation du code

## 9 Jeu du morpion

Un autre groupe d'élèves souhaite réaliser un jeu de morpion. C'est un jeu entre deux joueurs dans lequel chaque joueur ou joueuse inscrit, l'un après l'autre, son symbole sur une grille. Le premier joueur ou la première joueuse qui parvient à aligner trois de ses symboles horizontalement, verticalement ou en diagonale gagne le jeu. Le cahier des charges établi avec les élèves est le suivant :

- Le jeu doit être muni d'une interface graphique qui doit être programmée en langage Python.
- Le jeu est conçu de telle sorte que le joueur se confronte à l'ordinateur qui jouerait de manière optimale, en se reposant sur l'algorithme min-max décrit dans le document 21.

41. Proposer une activité qui vous permettrait de présenter aux élèves d'une classe de terminale NSI les concepts d'arbre binaire, de parcours en profondeur et en largeur de cet arbre. On pourra se référer à l'extrait du programme fourni dans le document 18.

On peut introduire la notion d'arbre binaire par des exemples. Par exemple, on peut modéliser le déroulement d'une partie du jeu de société "Qui est-ce ?", par un ensemble de questions qui n'appellent que des réponses de type oui - non et dont la réponse à chaque question détermine la question suivante par un arbre binaire.

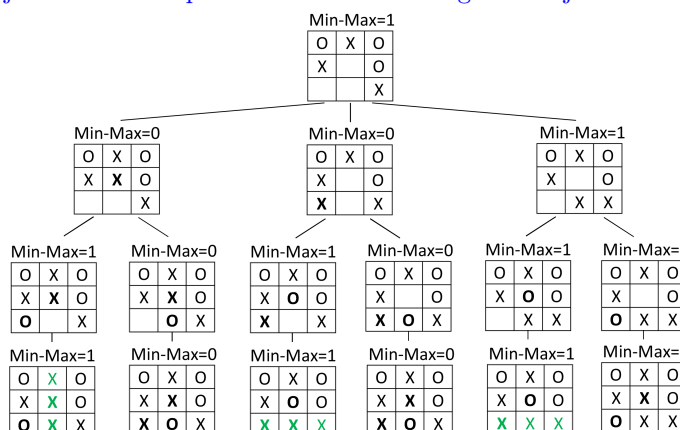
On peut donner alors quelques définitions : nœud, racine, nœud parent, nœud fils, feuille, arête, sous-arbre, taille, profondeur d'un nœud, hauteur... Dans le cas du jeu "Qui est-ce ?", on peut mettre en évidence deux stratégies extrêmes : l'une (optimale) permettant de scinder chaque groupe d'individus en deux groupes égaux, l'autre (nulle) ne permettant pas d'isoler d'individus. Ces deux stratégies permettant d'introduire les concepts d'arbre filiforme et d'arbre complet. À cette occasion, on pourra introduire l'encadrement de la hauteur d'un arbre binaire :  $\lfloor \log_2(n) \rfloor \leq h \leq n - 1$ . On pourra aussi faire remarquer aux élèves que les arbres sont des structures récursives.

Les algorithmes sur les arbres binaires pourront être abordés dans un deuxième temps : calcul de la hauteur, parcours en largeur, parcours en profondeur (infixe, suffixe et préfixe)...

On peut demander aux élèves d'implémenter ces algorithmes en Python à partir des algorithmes en langage naturel (cela sera l'occasion d'aborder l'implémentation de la structure arbre en Python).

42. Expliquer le fonctionnement de l'algorithme min-max comme vous le feriez auprès des élèves du groupe ayant choisi ce projet.

L'objectif de l'algorithme du min-max est d'attribuer un score à chaque situation de jeu, selon qu'elle conduise à une victoire de l'IA (score = 1), à sa défaite (score = -1) ou encore à un cas d'égalité (score = 0), en admettant que l'adversaire humain joue de manière optimale. Pour illustrer son fonctionnement auprès d'élèves, on peut leur proposer de construire l'arbre des situations de jeu ci-dessous en fournissant la situation initiale située à sa racine. L'IA place les croix « X », le joueur humain place les « O » sur la grille de jeu. C'est au tour de l'IA de jouer.



On peut ensuite demander aux élèves de calculer le score retourné par la fonction min-max pour chaque situation de jeu figurant dans cet arbre en leur indiquant que :

- Chaque situation de jeu finale (feuilles de l'arbre) se voit attribuer le score 1, -1 ou 0. Ce score est calculé dans le cas de base de la fonction min-max et est calculé de la manière suivante.

Valeur de retour de « coefficient »	Valeur de retour de « évaluer »	Valeur de retour de « min-max »	Interprétation
1 = l'IA	1 = Gagne	1	L'IA gagne
1 = l'IA	0 = Match nul	0	Match nul
1 = l'IA	-1 = Perd	-1	L'IA perd
-1 = l'humain	1 = Gagne	-1	L'IA perd
-1 = l'humain	0 = Match nul	0	Match nul
-1 = l'humain	-1 = Perd	1	L'IA gagne

- Dans le cas où le tour suivant est joué par l'IA, le score retenu par la fonction min-max est le maximum des scores des situations de jeu atteignables au tour suivant. Par ce moyen, le score attribué tient compte d'un déroulement de jeu dans lequel les décisions prises par l'IA lui sont les plus favorables. A l'inverse, dans le cas où le tour suivant est joué par l'humain, le score retenu est le minimum des scores des situations de jeu atteignables au tour suivant.
43. Préciser pourquoi il répond à la problématique d'un jeu optimal de la part de l'ordinateur, et pourquoi il se fait en temps de calcul raisonnable dans le cas du jeu de morpion.

Le fonctionnement de l'algorithme expliqué à la question précédente met en évidence que l'IA joue les coups qui lui sont le plus favorable, en anticipant que l'humain qui l'oppose jouera les coups les plus préjudiciables. L'algorithme répond ainsi à la problématique d'un jeu optimal. Le temps de calcul reste raisonnable dans le cas du morpion, car l'arbre construit possède 10 niveaux correspondant à l'état initial du jeu et aux 9 tours joués. Chaque nœud  $S$  de cet arbre représente une situation de jeu, et donc possède au plus  $n - 1$  fils (excepté pour le nœud de profondeur nulle, qui a 9 fils) où  $n$  est le nombre de nœuds de même parent que  $S$ . Cela s'explique par le fait qu'à chaque tour de jeu, le nombre de case disponible est décrémenté. En conséquence, un majorant du nombre de situations de jeu à évaluer est  $9! = 362880$ . L'ordre de grandeur obtenu est tout à fait envisageable.

44. Proposer trois tests permettant de vérifier le bon fonctionnement de l'application réalisée par les élèves au cours de ce projet.

Les tests suivant peuvent être menés pour contrôler le bon fonctionnement de l'application :

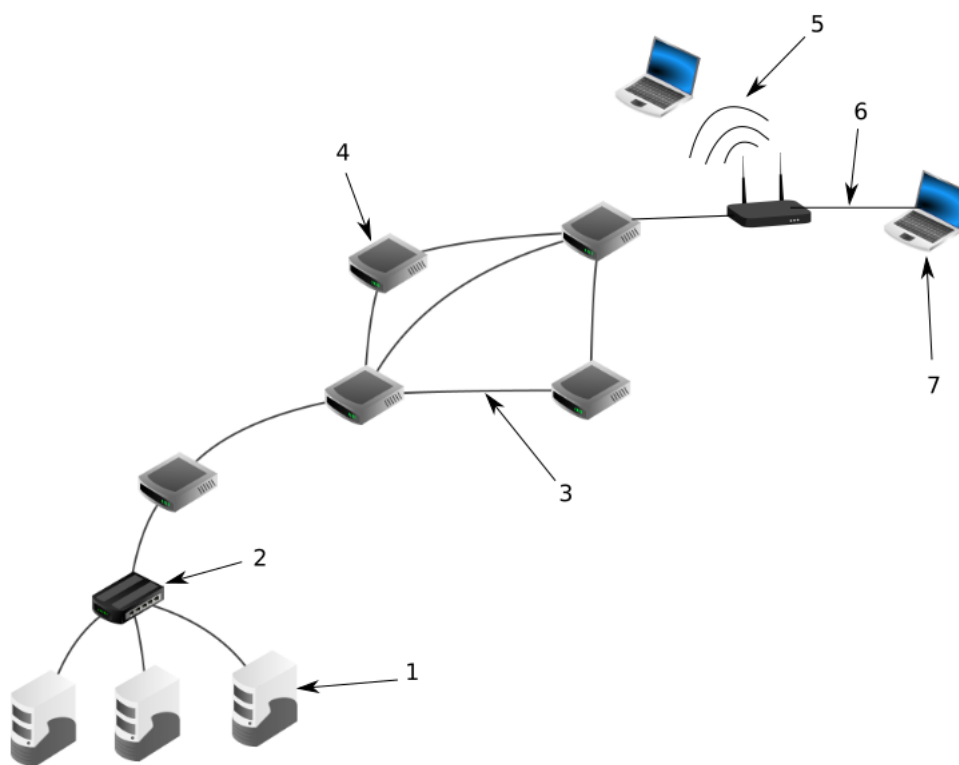
- Test des mécanismes de base du jeu : impossibilité de jouer 2 fois sur la même case, détection de la victoire de l'humain ou de l'IA, ...
- Le joueur humain joue de façon à laisser volontairement des opportunités de victoire à l'IA. On vérifie dans ce cas de figure que l'IA saisit ces opportunités et remporte le jeu.
- Le joueur humain joue de manière optimale afin de vérifier que toutes les parties se terminent bien par un match nul.

45. Proposer une grille d'évaluation pour ce projet.

Ce projet est demande de solides compétences en algorithmique pour des élèves de terminale NSI. A ce titre, il est envisageable d'évaluer le bon fonctionnement des éléments décisionnels du logiciel, plutôt que sur son rendu (graphique ou interactions HM). Parmi ces éléments, on peut retenir :

- Le bon fonctionnement de la fonction d'évaluation d'une situation.
- Le bon fonctionnement de la fonction d'inventaire des évaluations.
- La construction d'un appel récursif.
- La transmission des paramètres aux différentes fonctions de décision du jeu.
- Le choix de configurations de jeu pertinentes pour tester le bon fonctionnement des fonctions de décision du jeu.
- L'ajout de fonctions de débogage (ex : sorties en console) pour contrôler les états de jeu traités ainsi que les décisions retenues.
- ...

## Annexes exercice 1



DOCUMENT 7 – Schéma réseau

Contenus	Capacités attendues
Protocole TCP/IP : paquets, routage des paquets	Distinguer le rôle des protocoles IP et TCP. Caractériser les principes du routage et ses limites. Distinguer la fiabilité de transmission et l'absence de garantie temporelle.
Adresses symboliques et serveurs DNS	Sur des exemples réels, retrouver une adresse IP à partir d'une adresse symbolique et inversement.
Réseaux pair-à-pair	Décrire l'intérêt des réseaux pair-à-pair ainsi que les usages illicites qu'on peut en faire.
Indépendance d'internet par rapport au réseau physique	Caractériser quelques types de réseaux physiques : obsolètes ou actuels, rapides ou lents, filaires ou non. Caractériser l'ordre de grandeur du trafic de données sur internet et son évolution.

DOCUMENT 8 – Extrait programme de SNT (thème "Internet")

Contenus	Capacités attendues
Réseaux sociaux existants	Distinguer plusieurs réseaux sociaux selon leurs caractéristiques, y compris un ordre de grandeur de leurs nombres d'abonnés. Paramétrer des abonnements pour assurer la confidentialité de données personnelles.
Modèle économique des réseaux sociaux	Identifier les sources de revenus des entreprises de réseautage social.
Rayon, diamètre et centre d'un graphe	Déterminer ces caractéristiques sur des graphes simples.
Notion de « petit monde » Expérience de Milgram	Décrire comment l'information présentée par les réseaux sociaux est conditionnée par le choix préalable de ses amis.
Harcèlement numérique	Connaître les dispositions de l'article 222-33-2-2 du code pénal.

DOCUMENT 9 – Extrait programme de SNT (thème "Les réseaux sociaux")

Contenus	Capacités attendues
Repères historiques	Définir les étapes du développement du Web.
Hypertexte	Maîtriser les renvois d'un texte à différents contenus.
Langages HTML et CSS	Distinguer ce qui relève du contenu d'une page et de son style de présentation. Étudier et modifier une page HTML simple.
URL	Décomposer l'URL d'une page. Reconnaître les pages sécurisées.
Requête HTTP	Décomposer le contenu d'une requête HTTP et identifier les paramètres passés.
Modèle client/serveur	Inspecter le code d'une page hébergée par un serveur et distinguer ce qui est exécuté par le client et par le serveur.
Moteurs de recherche : principes et usages	Mener une analyse critique des résultats fournis par un moteur de recherche. Comprendre que toute requête laisse des traces.
Paramètres de sécurité d'un navigateur	Maîtriser les réglages les plus importants concernant la gestion des cookies, la sécurité et la confidentialité d'un navigateur.

DOCUMENT 10 – Extrait programme de SNT (thème "Le Web")

Contenus	Capacités attendues	Commentaires
Modèle d'architecture séquentielle (von Neumann)	Distinguer les rôles et les caractéristiques des différents constituants d'une machine. Dérouler l'exécution d'une séquence d'instructions simples du type langage machine.	La présentation se limite aux concepts généraux. On distingue les architectures monoprocesseur et les architectures multiprocesseur. Des activités débranchées sont proposées. Les circuits combinatoires réalisent des fonctions booléennes.
Transmission de données dans un réseau Protocoles de communication Architecture d'un réseau	Mettre en évidence l'intérêt du découpage des données en paquets et de leur encapsulation. Dérouler le fonctionnement d'un protocole simple de récupération de perte de paquets (bit alterné). Simuler ou mettre en œuvre un réseau.	Le protocole peut être expliqué et simulé en mode débranché. Le lien est fait avec ce qui a été vu en classe de seconde sur le protocole TCP/IP. Le rôle des différents constituants du réseau local de l'établissement est présenté.

DOCUMENT 11 – Extrait du programme de première de la spécialité NSI (thème "Architectures matérielles et systèmes d'exploitation")



Contenus	Capacités attendues	Commentaires
Modalités de l'interaction entre l'homme et la machine  Événements	Identifier les différents composants graphiques permettant d'interagir avec une application Web. Identifier les événements que les fonctions associées aux différents composants graphiques sont capables de traiter.	Il s'agit d'examiner le code HTML d'une page comprenant des composants graphiques et de distinguer ce qui relève de la description des composants graphiques en HTML de leur comportement (réaction aux événements) programmé par exemple en JavaScript.
Interaction avec l'utilisateur dans une page Web	Analyser et modifier les méthodes exécutées lors d'un clic sur un bouton d'une page Web.	
Interaction client-serveur. Requêtes HTTP, réponses du serveur	Distinguer ce qui est exécuté sur le client ou sur le serveur et dans quel ordre. Distinguer ce qui est mémorisé dans le client et retransmis au serveur. Reconnaître quand et pourquoi la transmission est chiffrée.	Il s'agit de faire le lien avec ce qui a été vu en classe de seconde et d'expliquer comment on peut passer des paramètres à un site grâce au protocole HTTP.
Formulaire d'une page Web	Analyser le fonctionnement d'un formulaire simple. Distinguer les transmissions de paramètres par les requêtes POST ou GET.	Discuter les deux types de requêtes selon le type des valeurs à transmettre et/ou leur confidentialité.

DOCUMENT 12 – Extrait du programme de première de la spécialité NSI (thème "Interactions entre l'homme et la machine sur le Web")

Contenus	Capacités attendues	Commentaires
Composants intégrés d'un système sur puce.	Identifier les principaux composants sur un schéma de circuit et les avantages de leur intégration en termes de vitesse et de consommation.	Le circuit d'un téléphone peut être pris comme un exemple : microprocesseurs, mémoires locales, interfaces radio et filaires, gestion d'énergie, contrôleurs vidéo, accélérateur graphique, réseaux sur puce, etc.
Gestion des processus et des ressources par un système d'exploitation.	Décrire la création d'un processus, l'ordonnement de plusieurs processus par le système. Mettre en évidence le risque de l'interblocage ( <i>deadlock</i> ).	À l'aide d'outils standard, il s'agit d'observer les processus actifs ou en attente sur une machine. Une présentation débranchée de l'interblocage peut être proposée.
Protocoles de routage.	Identifier, suivant le protocole de routage utilisé, la route empruntée par un paquet.	En mode débranché, les tables de routage étant données, on se réfère au nombre de sauts (protocole RIP) ou au coût des routes (protocole OSPF). Le lien avec les algorithmes de recherche de chemin sur un graphe est mis en évidence.
Sécurisation des communications.	Décrire les principes de chiffrement symétrique (clef partagée) et asymétrique (avec clef privée/clef publique). Décrire l'échange d'une clef symétrique en utilisant un protocole asymétrique pour sécuriser une communication HTTPS.	Les protocoles symétriques et asymétriques peuvent être illustrés en mode débranché, éventuellement avec description d'un chiffrement particulier. La négociation de la méthode de chiffrement du protocole SSL ( <i>Secure Sockets Layer</i> ) n'est pas abordée.

DOCUMENT 13 – Extrait programme de terminale NSI (thème "Architectures matérielles, systèmes d'exploitation et réseaux")



## Démarche de projet

Un enseignement d'informatique ne saurait se réduire à une présentation de concepts ou de méthodes sans permettre aux élèves de se les approprier en développant des projets.

Un quart au moins de l'horaire total de la spécialité est réservé à la conception et à l'élaboration de projets conduits par les élèves.

Les projets réalisés par les élèves, sous la conduite du professeur, constituent un apprentissage fondamental tant pour l'appropriation des concepts informatiques que pour l'acquisition de compétences. En classe de première comme en classe terminale, ils peuvent porter sur des problématiques issues d'autres disciplines et ont essentiellement pour but d'imaginer des solutions répondant à un problème ; dans la mesure du possible, il convient de laisser le choix du thème du projet aux élèves. Il peut s'agir d'un approfondissement théorique des concepts étudiés en commun, d'une application à d'autres disciplines telle qu'une simulation d'expérience, d'exploitation de modules liés à l'intelligence artificielle et en particulier à l'apprentissage automatique, d'un travail sur des données socioéconomiques, du développement d'un logiciel de lexicographie, d'un projet autour d'un objet connecté ou d'un robot, de la conception d'une bibliothèque implémentant une structure de données complexe, d'un problème de traitement d'image ou de son, d'une application mobile, par exemple de réalité virtuelle ou augmentée, du développement d'un site *Web* associé à l'utilisation d'une base de données, de la réalisation d'un interpréteur d'un mini-langage, de la recherche d'itinéraire sur une carte (algorithme A\*), d'un programme de jeu de stratégie, etc.

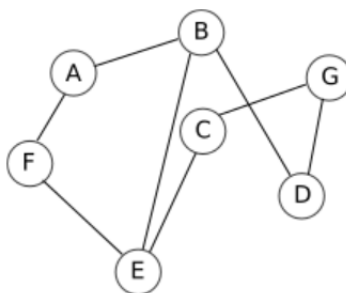
La conduite d'un projet inclut des points d'étape pour faire un bilan avec le professeur, valider des éléments, contrôler l'avancement du projet ou en adapter les objectifs, voire le redéfinir partiellement, afin de maintenir la motivation des élèves.

Les professeurs veillent à ce que les projets restent d'une ambition raisonnable afin de leur permettre d'aboutir.

### DOCUMENT 14 – Extrait programme de terminale NSI

#### Exercice III (5 points)

Soit le graphe suivant :



1. Donnez une implémentation de ce graphe en langage Python.
2. En partant du sommet A, donnez la liste des sommets parcourus dans le cas d'un parcours « en profondeur d'abord » (attention à l'ordre des sommets).
3. En partant du sommet A, donnez la liste des sommets parcourus dans le cas d'un parcours « en largeur d'abord » (attention à l'ordre des sommets).
4. Écrivez l'algorithme permettant de réaliser un parcours « en largeur d'abord »

### DOCUMENT 15 – Exercice proposé à des élèves de terminale en spécialité NSI

Contenus	Capacités attendues	Commentaires
Algorithmes sur les arbres binaires et sur les arbres binaires de recherche.	Calculer la taille et la hauteur d'un arbre. Parcourir un arbre de différentes façons (ordres infixe, préfixe ou suffixe ; ordre en largeur d'abord). Rechercher une clé dans un arbre de recherche, insérer une clé.	Une structure de données récursive adaptée est utilisée. L'exemple des arbres permet d'illustrer la programmation par classe. La recherche dans un arbre de recherche équilibré est de coût logarithmique.
Algorithmes sur les graphes.	Parcourir un graphe en profondeur d'abord, en largeur d'abord. Repérer la présence d'un cycle dans un graphe. Chercher un chemin dans un graphe.	Le parcours d'un labyrinthe et le routage dans Internet sont des exemples d'algorithme sur les graphes. L'exemple des graphes permet d'illustrer l'utilisation des classes en programmation.

### DOCUMENT 16 – Extrait programme de terminale NSI (thème "Algorithmique")

### Exercice III

- 1) On peut utiliser un dictionnaire pour implémenter le graphe en python.

$G = \{ 'A': ['F', 'B'], 'B': ['E', 'D'], 'C': ['E', 'G'], 'D': ['G'], 'E': ['F'] \}$

- 2) Parcours en profondeur d'abord:

$A \rightarrow F \rightarrow E \rightarrow C \rightarrow G \rightarrow D \rightarrow B$

- 3) Parcours en largeur d'abord

$A \rightarrow B \rightarrow E \rightarrow F \rightarrow D \rightarrow G \rightarrow C$

4)

VARIABLE

G: graphe

s: nœud origine

u: nœud

v: nœud

p: pile (initialement vide)

DEBUT

s.couleur ← noir

piles(s, p)

tant que p non vide :

u ← depiler(p)

pour chaque sommet v adj à u :

si v.couleur n'est pas noir :

v.couleur ← noir

piles(v, p)

fin si  
fin pour  
fin tant que

## Structures de données

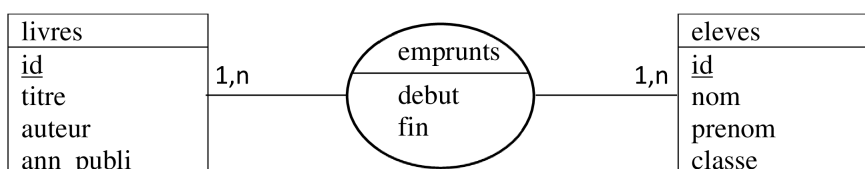
L'écriture sur des exemples simples de plusieurs implémentations d'une même structure de données permet de faire émerger les notions d'interface et d'implémentation, ou encore de structure de données abstraite.

Le paradigme de la programmation objet peut être utilisé pour réaliser des implémentations effectives des structures de données, même si ce n'est pas la seule façon de procéder.

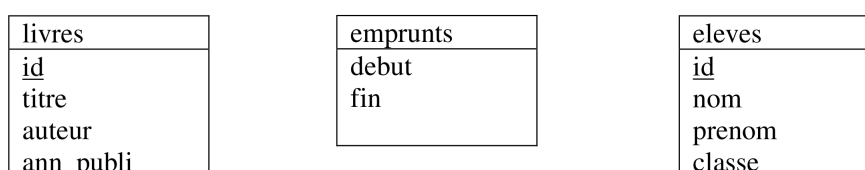
Le lien est établi avec la notion de modularité qui figure dans la rubrique « langages et programmation » en mettant en évidence l'intérêt d'utiliser des bibliothèques ou des API (*Application Programming Interface*).

Contenus	Capacités attendues	Commentaires
Structures de données, interface et implémentation.	Spécifier une structure de données par son interface. Distinguer interface et implémentation. Écrire plusieurs implémentations d'une même structure de données.	L'abstraction des structures de données est introduite après plusieurs implémentations d'une structure simple comme la file (avec un tableau ou avec deux piles).
Vocabulaire de la programmation objet : classes, attributs, méthodes, objets.	Écrire la définition d'une classe. Accéder aux attributs et méthodes d'une classe.	On n'aborde pas ici tous les aspects de la programmation objet comme le polymorphisme et l'héritage.
Listes, piles, files : structures linéaires. Dictionnaires, index et clé.	Distinguer des structures par le jeu des méthodes qui les caractérisent. Choisir une structure de données adaptée à la situation à modéliser. Distinguer la recherche d'une valeur dans une liste et dans un dictionnaire.	On distingue les modes FIFO ( <i>first in first out</i> ) et LIFO ( <i>last in first out</i> ) des piles et des files.
Arbres : structures hiérarchiques. Arbres binaires : nœuds, racines, feuilles, sous-arbres gauches, sous-arbres droits.	Identifier des situations nécessitant une structure de données arborescente. Évaluer quelques mesures des arbres binaires (taille, encadrement de la hauteur, etc.).	On fait le lien avec la rubrique « algorithmique ».
Graphes : structures relationnelles. Sommets, arcs, arêtes, graphes orientés ou non orientés.	Modéliser des situations sous forme de graphes. Écrire les implémentations correspondantes d'un graphe : matrice d'adjacence, liste de successeurs/de prédécesseurs. Passer d'une représentation à une autre.	On s'appuie sur des exemples comme le réseau routier, le réseau électrique, Internet, les réseaux sociaux. Le choix de la représentation dépend du traitement qu'on veut mettre en place : on fait le lien avec la rubrique « algorithmique ».

DOCUMENT 18 – Extrait du programme de terminale de la spécialité NSI (thème "Structures de données")



DOCUMENT 19 – Modèle conceptuel de données retenu



DOCUMENT 20 – Tables implémentées par un élève

Lorsque l'IA doit jouer dans une situation de jeu donnée, son action est déterminée par le choix d'une nouvelle situation, prise au hasard, parmi toutes les situations de jeu atteignables dont l'évaluation par la fonction **min-max**, fournie ci-dessous est maximale. Dans cette fonction, on considérera que :

- La fonction **évaluer** retourne un entier 1 = gagnant, 0 = match nul ou -1 = perdant selon que la situation de fin de jeu fournie en argument est favorable, ne départage pas ou est défavorable au joueur fourni en argument.
- La fonction **coefficient** retourne un entier 1 = IA ou -1 = humain, selon le joueur fourni en argument.
- La fonction **situations-atteignables** retourne, pour une situation de jeu fournie en argument, l'ensemble des situations pouvant être atteintes au tour de jeu suivant.
- Les fonctions **min** et **max** retournent les minimum et maximum des ensembles de valeurs qui leur sont fournies en argument.

1. **fonction min-max**(situation, joueur)
2.     si situation est finale
3.         retourner **évaluer**(situation, joueur)  $\times$  **coefficient**(joueur)
4.     sinon
5.         situationsSuivantes  $\leftarrow$  **situations-atteignables**(situation)
6.         si joueur = IA
7.             retourner **max**{**min-max**(suivante, humain) pour suivante  $\in$  situationsSuivantes}
8.         sinon
9.             retourner **min**{**min-max**(suivante, IA) pour suivante  $\in$  situationsSuivantes}
10.        fin si
11.     fin si
12. fin fonction

DOCUMENT 21 – Principe général du fonctionnement de l'algorithme du min-max