Les six parties qui suivent sont indépendantes.

I Autour de la complexité

▸ **Question 1.** *Un élève a du mal à voir la différence entre une fonction* `Python` *et un algorithme, aidez-le à y voir plus clair.*

▸ **Question 2.** *Une élève vous demande si les fonctions* `Python` *ont quelque chose à voir avec les fonctions en mathématiques, que lui répondez-vous ?*

▸ **Question 3.** *Proposer une activité permettant de mettre en valeur les différentes complexités temporelles classiques, après avoir défini ces dernières (il y en a trois). Quelles difficultés les élèves pourraient rencontrer sur celle-ci ?*

▸ **Question 4.** *Un élève vous demande pourquoi on parle de complexité* asymptotique*, que lui répondez vous ?*

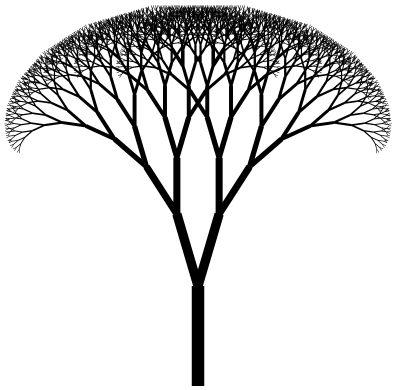
▸ **Question 5.** *Une élève vous demande si on peut toujours associer à un algorithme une complexité temporelle, que lui répondez vous ?*

II Autour de la récursivité

▸ **Question 1.** *Un élève vous demande si tout algorithme récursif admet un algorithme itératif équivalent, que lui répondez vous ?*

▸ **Question 2.** *Proposez un tp autour de la récursivité pour la rotation d'une image bitmap de 90 degrés. Vous préciserez les bibliothèques* `Python` *utilisées, les difficultés attendues (et comment la structure de votre TP permet de guider les élèves sur celles-ci) et une rapide correction.*

▸ **Question 3.** *Proposez un tp autour de la récursivité pour les arbres fractales. Vous préciserez les bibliothèques* `Python` *utilisées, les difficultés attendues, de possibles extensions de l'activité pour les élèves les plus rapides et une rapide correction.*



▸ **Question 4.** *Proposez un tp autour du problème des huits dames à la manière des questions précédentes. L'objectif de ce problème est de placer huit dames sur une échiquier de 64 cases de sorte qu'elles ne se menacent pas les unes les autres. Vous lierez l'algorithme de résolution esquissé à l'exploration maligne d'un arbre bien choisi.*

III Autour de la représentation des types de base

▸ **Question 1.** *Un élève vous demande de faire un point sur le log en informatique, et si celui-ci a un lien avec le* ln *vu en cours de mathématiques, que lui répondez-vous ?*

Dans la suite, on considère la représentation d'un entier naturel $n \in [\![0, 255]\!]$ en mémoire: le fragment mémoire utilisé pour stocker l'entier est vu comme tableau de booléens.

▸ **Question 2.** *Quelle est la taille du tableau nécessaire ? Proposer deux fonctions* `convert_to_int` *et* `convert_to_bin` *permettant de convertir un entier naturel en sa représentation binaire et inversement, vous mettrez en valeur la convention que vous utilisez.*

▸ **Question 3.** *On souhaite tester si un nombre binaire est zéro. Guidez les élèves vers la proposition d'une fonction lazy.*

▸ **Question 4.** *Rappelez l'algorithme d'addition sur les entiers naturels considérés ici. Justifier rapidement qu'il peut s'exprimer sous la forme d'un circuit logique.*

On considère maintenant un entier relatif $z \in [\![-511, 511]\!]$.

▸ **Question 5.** *Comment pourrait-on naïvement représenter cet entier relatif ? Montrez que cette représentation n'est pas compatible avec l'algorithme d'addition proposé précédemment. En réalité, quelle est la représentation utilisée en pratique pour les entiers relatifs ?*

IV Correction de code

▸ **Question 1.** *Pour chaque production d'une ou d'un élève en réponse à une question posée, identifier les points faibles de l'élève et des façons d'y remédier (questions intermédiaires, point de cours, activitée débranchée, . . .).*

```
def est_voisin(G, i, j):
    if j in G[i]:
        return True
    else:
        return False
##########################################################################
coord_A = (3, 2)
coord_B = (6, 7)
def vecteur(coord_depart, coord_arrivee):
    return coord_arrivee - coord_départ
##########################################################################
def triFusion_fusion(liste_A, liste_B):
    if liste_A[1] < liste_B[1]:
        return liste_A[1] + triFusion_fusion(liste_A[2:], liste_B)
    elif liste_A[1] > liste_B[1]:
        return liste_A[1] + triFusion_fusion(liste_A, liste_B[2:])
    elif liste_A == []: # liste_B == [] aussi
        return []
##########################################################################
def racineCubique(n):
    return sqrt(n/3)
assert(racineCubique(27) == 3)
##########################################################################
def f(x = lambda n: n%2, y):
    z = True
    for t in y:
        z &= x(t)
    return z
##########################################################################
class Vecteur(x,y):
    def square_norm():
        return x^2 + y^2
##########################################################################
def split_IP(ip_bin): # Si ip=128, ip_bin=010000000
    mask_bin = 11111111.11111111.11111111.00000000 # mask = 255.255.255.0
    network_part = (ip_bin ^ mask_bin) >> 8
    device_part = ip_bin % (1 << 8)
    return (network_part, device_part)
```

V Notation polonaise inverse et évaluation d'arbre

La *notation polonaise inverse* permet d'écrire de façon non ambigüe les formules arithmétiques sans utiliser de parenthèses. Dans cette dernière, les opérandes sont positionnées avant leur opérateur : on écrira `4 5 +` à la place de `4 + 5`. La NPI s'appuie sur une structure implicite de pile : les opérandes et les résultats de calculs sont ajoutés à la pile, les calculs sont effectués sur les premiers éléments dépilés (selon l'arité de l'opérateur considéré). Par exemple, `4 5 + 10 -` vaut `1`.

▸ **Question 1.** *Que vaut `2 3 5 + 7 * -` ?*

▸ **Question 2.** *Compléter le code suivant, qui permet d'évaluer une expression arithmétique en NPI pour quelques opérateurs d'arité 2. Comment s'appelle le style de programmation utilisé ici ?*

```
def eval_npi(expr: str) -> int:
    stack = [] # Recall that list-based stack use .pop() and .append() operators

    dict_op = {'+': # Compléter ici,
               '-': # Compléter ici,
               '*': # Compléter ici}

    for val in expr.split(' '):
        if val.isdigit():
            # Compléter ici
        else:
            # Compléter ici
    # Compléter ici
```

▸ **Question 3.** *Des élèves souhaitent étendre ce code à des opérateurs d'arités diverses mais ne savent pas par où commencer, comment les aiguillerez vous ?*

▸ **Question 4.** *On souhaite maintenant écrire une fonction `build_tree_from_npi` qui construit l'arbre associé à l'expression donnée en entrée. À quoi ressemble cet arbre ? Expliquer (pas besoin de coder) comment adapter `eval_npi` (si vous avez besoin d'introduire des structures de données particulières, détaillez les).*

▸ **Question 5.** *Donner un algorithme evalTree qui évalue l'arbre d'expression arithmétique donné en entrée. Quel parcours avez vous utilisé ?*

▸ **Question 6.** *Faire un lien entre cet exercice et la compilation de programme.*

VI Simulation de systèmes distribués en graphe

▸ **Question 1.** *Une élève vous demande si l'algorithme de Dijkstra se rapproche plus du parcours en largeur ou en profondeur, que lui répondez vous après avoir rappelé l'algorithme en question.*

▸ **Question 2.** *Écrire deux fonctions pythons* `convertGraph_toMatrix` *et* `convertGraph_toAdjLists` *qui permettent de passer d'une représentation de graphe à une autre. Vous utiliserez des listes en compréhension. On supposera que les sommets sont labellés par l'ensemble* $[\![0, n-1]\!]$.

Dans la suite, on s'intéresse à modéliser le protocole de routage RIP pour une activité que vous proposeriez à vos élèves. On se place dans le cas simplifié d'un réseau fixe, sans panne et qui dispose d'une horloge commune.

▸ **Question 3.** *Rappeler, dans les grandes lignes, comment fonctionne le protocole RIP – pour Routing Information Protocol – (phase d'initialisation et transfert de paquet).*

▸ **Question 4.** *On suppose que le programme exécuté en boucle sur un routeur est de la forme "ReadInbox-LocalComputation-OutputMessages". Proposer une modélisation* `Python` *d'un petit réseau de routeurs capables de communiquer entre eux, et une fonction* `simulation(N)` *capable de simuler l'évolution de ce réseau pendant* `N` *cycles.*

▸ **Question 5.** *Proposer le plan d'une activité autour de cette simulation telle que vous la proposeriez à vos élèves : support donné, questions posées, difficultés envisagées et indications associées, extensions possibles...*