

Échelle d'évaluation standard : **Notation en attente du traitement des éventuelles demandes de précision**

Échelle d'évaluation pondérée : **Notation en attente du traitement des éventuelles demandes de précision**

Algorithmique avancée : Algorithmique et complexité (CCTL)

Échelle d'évaluation standard : **Notation en attente du traitement des éventuelles demandes de précision**

Question 1Question d'association

Mettez dans l'ordre de temps de calcul asymptotiquement croissant les ordres de grandeur de complexité algorithmique suivantes :

a : $O(n^2)$
b : $O(n)$
c : $O(2^n)$
d : $O(n^3)$
e : $O(n!)$
f : $O(\log(n))$

Réponses correctes0 discordance

Élément à associer	Réponse attendue	Réponse saisie	Réponse discordante
1	f	f	Non
2	b	b	Non
3	a	a	Non
4	d	d	Non
5	c	c	Non
6	e	e	Non

Question 2Question à réponse unique

Complétez :
Si un algorithme a une complexité temporelle asymptotique $O(g(n))$...

Réponses correctes

	Réponse attendue	Réponse saisie	Réponse discordante	
A	<input type="checkbox"/>	<input type="checkbox"/>	Non	Si la quantité de données est suffisamment grande, il faut au pire $c \times n$ opérations à l'algorithme pour résoudre une instance de taille $G(n)$, avec c une constante
B	<input type="checkbox"/>	<input type="checkbox"/>	Non	Si la quantité de données est suffisamment grande, il faut au mieux $c \times n$ opérations à l'algorithme pour résoudre une instance de taille $g(n)$, avec c une constante
C	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Non	Si la quantité de données est suffisamment grande, il faut $c \times g(n)$ opérations à l'algorithme résoudre une instance de taille n , avec c une constante
D	<input type="checkbox"/>	<input type="checkbox"/>	Non	Si la quantité de données est suffisamment petite, il faut au pire $c \times n$ opérations à l'algorithme pour résoudre une instance de taille $g(n)$, avec c une constante
E	<input type="checkbox"/>	<input type="checkbox"/>	Non	Si la quantité de données est suffisamment petite, il faut $c \times g(n)$ opérations à l'algorithme résoudre une instance de taille n , avec c une constante
F	<input type="checkbox"/>	<input type="checkbox"/>	Non	Si la quantité de données est suffisamment petite, il faut au mieux $c \times n$ opérations à l'algorithme pour résoudre une instance de taille $g(n)$, avec c une constante

Question 3

Question à réponses multiples

Que signifie $f(x) \in O(g(x))$?

Réponses correctes

0 discordance

	Réponse attendue	Réponse saisie	Réponse discordante	
A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Non	f est bornée par g à un facteur près pour un x suffisamment grand
B	<input type="checkbox"/>	<input type="checkbox"/>	Non	g est bornée par f à un facteur près pour un x suffisamment grand
C	<input type="checkbox"/>	<input type="checkbox"/>	Non	Il existe un algorithme de complexité f qui est bornée asymptotiquement par g
D	<input type="checkbox"/>	<input type="checkbox"/>	Non	Il existe un algorithme de complexité g qui est bornée asymptotiquement par f
E	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Non	Tout algorithme de complexité f peut être considéré comme un algorithme de complexité O(g(x))

Question 4

Question à réponse unique

On considère un algorithme calculant le degré maximal d'un graphe d'ordre n :

```
max = 0
Pour chaque sommet i du graphe :
  si max < degré(i)
    max = degré(i)
```

Dans une représentation par matrice d'adjacence, quelle est la complexité temporelle au pire de cet algorithme ?
L'opération degré(i) est réalisée en O(1)

Réponses correctes

	Réponse attendue	Réponse saisie	Réponse discordante	
A	<input type="checkbox"/>	<input type="checkbox"/>	Non	constante
B	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Non	linéaire
C	<input type="checkbox"/>	<input type="checkbox"/>	Non	logarithmique
D	<input type="checkbox"/>	<input type="checkbox"/>	Non	quadratique
E	<input type="checkbox"/>	<input type="checkbox"/>	Non	cubique
F	<input type="checkbox"/>	<input type="checkbox"/>	Non	exponentielle

Question 5

Question à réponses multiples

On considère la complexité temporelle d'un algorithme.
Sélectionnez l(es) affirmation(s) exacte(s)

Réponses correctes

0 discordance

	Réponse attendue	Réponse saisie	Réponse discordante	
A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Non	Un ordinateur ne peut pas exécuter un algorithme de complexité exponentielle
B	<input type="checkbox"/>	<input type="checkbox"/>	Non	Une complexité polynomiale est moins bonne qu'une complexité linéaire
C	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Non	La complexité temporelle permet d'évaluer le temps que met un algorithme pour se terminer
D	<input type="checkbox"/>	<input type="checkbox"/>	Non	Un algorithme de complexité quadratique est inexploitable
E	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Non	La complexité temporelle est exprimée de manière asymptotique

Question 6

Question d'association

En supposant que $P \neq NP$, placez les différentes catégories de classes de complexité aux différentes zones du schéma ci-dessous :

Réponses correctes

0 discordance

Élément à associer	Réponse attendue	Réponse saisie	Réponse discordante	
Zone 1	NP-Difficile	NP-Difficile	Non	
Zone 2	NP-Complet	NP-Complet	Non	
Zone 3	NP	NP	Non	
Zone 4	P	P	Non	

Question 7

Question à réponses multiples

Un problème est NP-Complet si et seulement si (2 réponses) :

Réponses correctes

0 discordance

	Réponse attendue	Réponse saisie	Réponse discordante	
A	<input type="checkbox"/>	<input type="checkbox"/>	Non	Il se résout en temps polynomial
B	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Non	Il appartient à NP
C	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Non	Il est au moins aussi difficile que tout problème de NP
D	<input type="checkbox"/>	<input type="checkbox"/>	Non	C'est un problème linéaire
E	<input type="checkbox"/>	<input type="checkbox"/>	Non	Il possède un certificat de complexité exponentielle

Question 8

Question à réponse unique

Soit le problème de la coupe maximale dans un graphe

Données : Un graphe G

Problème : Quel est le nombre maximal d'arêtes que l'on peut retirer de G sans qu'il devienne non connexe ?

Le problème de décision correspondant est :

Réponses correctes

	Réponse attendue	Réponse saisie	Réponse discordante	
A	<input type="checkbox"/>	<input type="checkbox"/>	Non	Étant donné un graphe G , est-il possible de retirer k arêtes de G sans le déconnecter ?
B	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Non	Étant donnés un graphe G et un entier k , est-il possible de retirer k arêtes sans déconnecter G ?
C	<input type="checkbox"/>	<input type="checkbox"/>	Non	Étant donné un graphe G , existe-t-il un entier k tel qu'il est possible de retirer k arêtes à G sans le déconnecter ?
D	<input type="checkbox"/>	<input type="checkbox"/>	Non	Étant donné un graphe G , quel est l'entier k tel qu'il est possible de retirer k arêtes sans déconnecter G ?
E	<input type="checkbox"/>	<input type="checkbox"/>	Non	Étant donnés un graphe G et un entier k , quelles sont les k arêtes que l'on peut retirer de G sans le déconnecter ?

Question 9

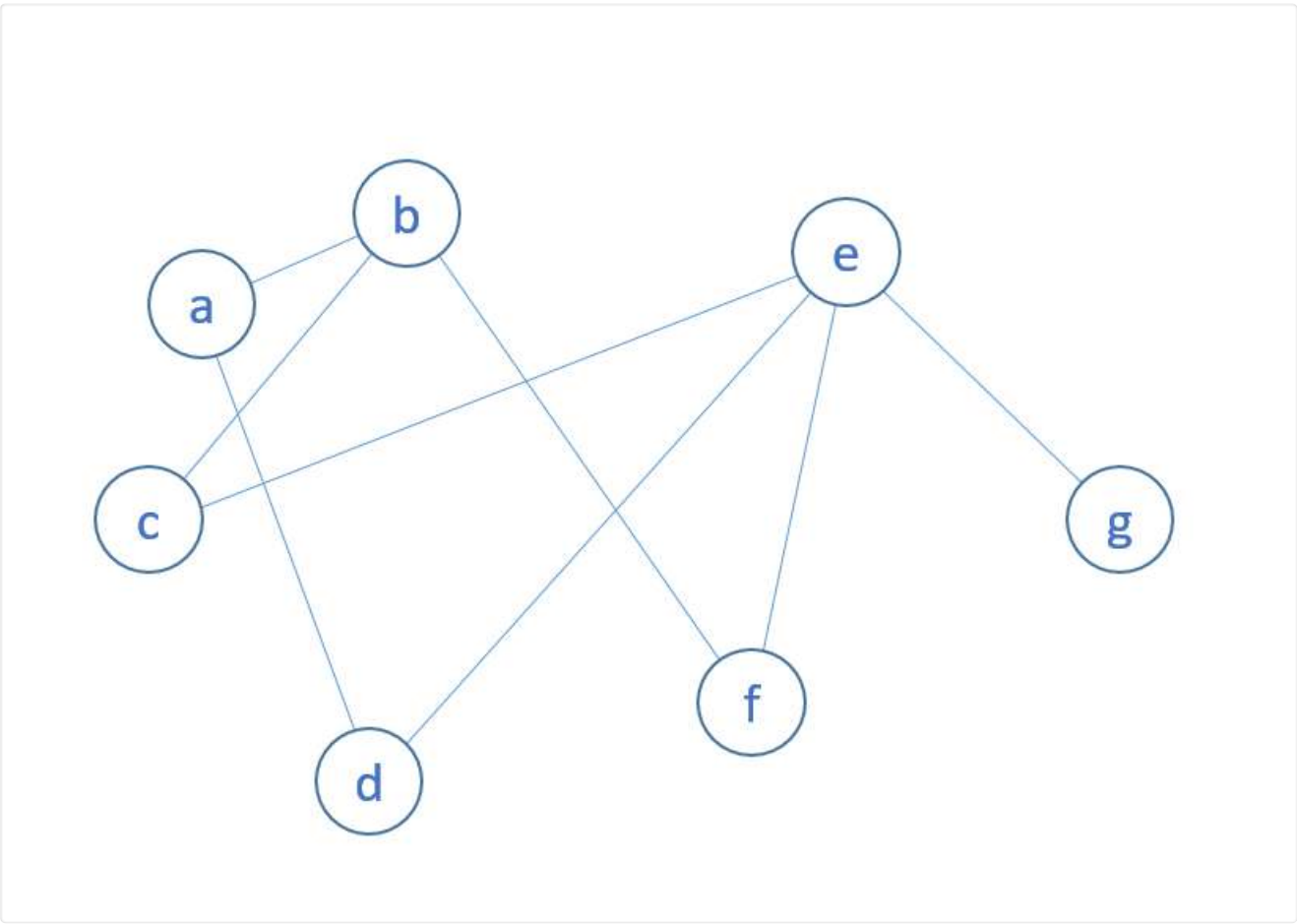
Question à réponse unique

On considère un problème d'optimisation P_{OPT} et le problème de décision P_{DEC} qui lui est associé. Quelle affirmation concernant les complexités de ces deux problèmes est vraie ?

Réponses correctes

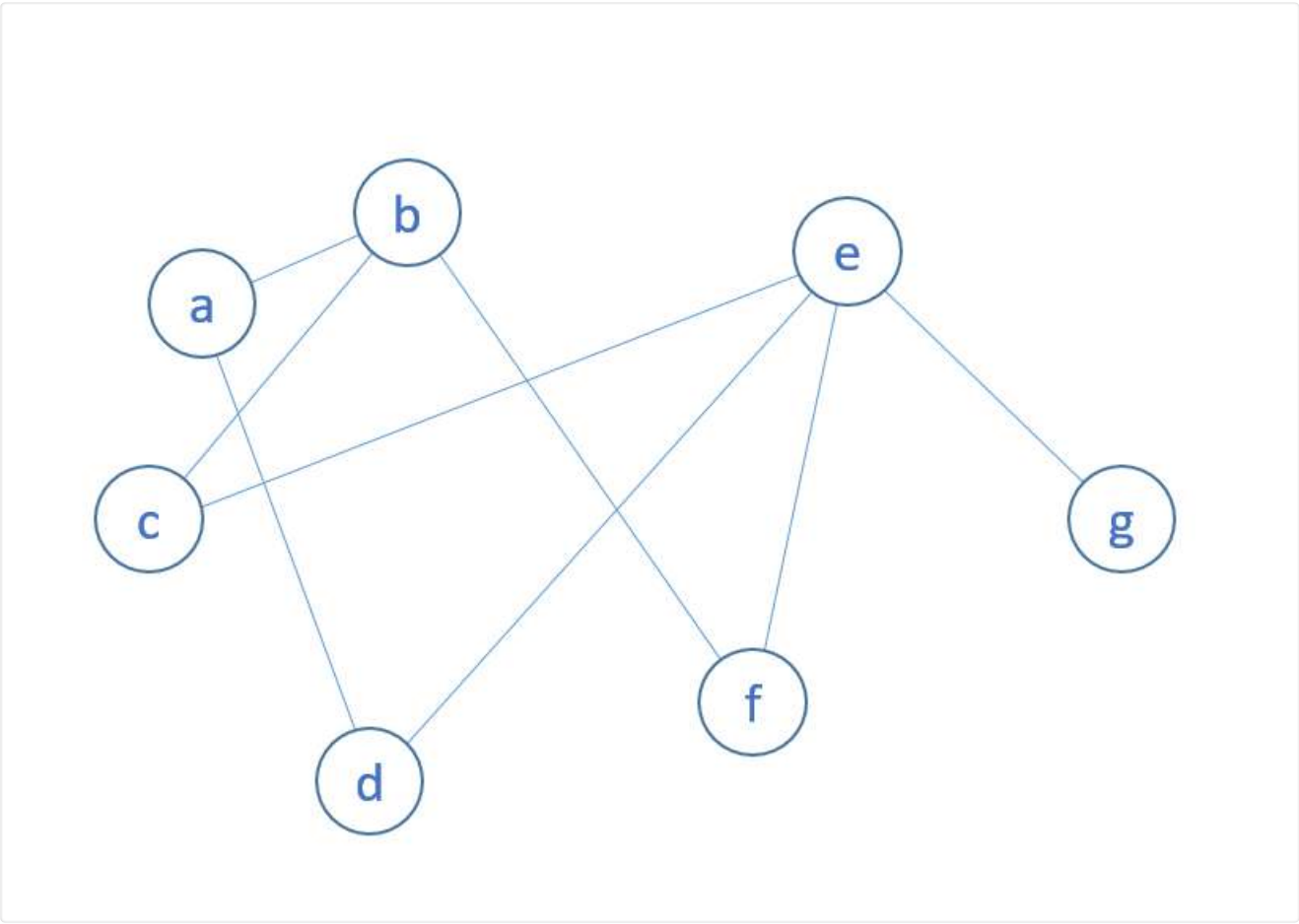
	Réponse attendue	Réponse saisie	Réponse discordante	
A	<input type="checkbox"/>	<input type="checkbox"/>	Non	$P_{\text{OPT}} > P_{\text{DEC}}$
B	<input type="checkbox"/>	<input type="checkbox"/>	Non	$P_{\text{OPT}} < P_{\text{DEC}}$
C	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Non	$P_{\text{OPT}} \geq P_{\text{DEC}}$
D	<input type="checkbox"/>	<input type="checkbox"/>	Non	$P_{\text{OPT}} \leq P_{\text{DEC}}$
E	<input type="checkbox"/>	<input type="checkbox"/>	Non	$P_{\text{OPT}} = P_{\text{DEC}}$

Quel est l'ordre du graphe suivant ?



Réponses correctes		0 discordance	
	Réponse attendue	Réponse saisie	Réponse discordante
	[7 ; 7]	7	Non
<div>● Commentaire de correction de la proposition</div>			

On considère le graphe suivant :



Indiquez si les propositions suivantes sont vraies ou fausses.

Réponses correctes		0 discordance	
Élément à associer	Réponse attendue	Réponse saisie	Réponse discordante
Il existe un cycle eulérien dans ce graphe.	faux	faux	Non
Il existe un chemin eulérien dans ce graphe.	vrai	vrai	Non

Question 12

Question à valeurs numériques

Combien de sommets possède le graphe représenté par la liste d’adjacence suivante, dont les numéros de sommet commencent à 1 ?

Head : 1, 2, 3, 6, 7, 7
Succ : 2, 1, 1, 3, 6, 3, 5

Réponses correctes

0 discordance

	Réponse attendue	Réponse saisie	Réponse discordante
	[5 ; 5]	5	Non

Commentaire de correction de la proposition

Question 13

Question à réponse ouverte et courte

On considère le graphe représenté par la liste d’adjacence suivante, dont les numéros de sommet commencent à 1

Head : 1, 2, 3, 6, 7, 8
Succ : 2, 1, 1, 3, 6, 3, 5

Quels sont les successeurs du sommet 3 ?

La réponse sera donnée sous forme de liste de numéros classés par ordre croissant, un espace entre chaque nombre.
Exemple : 1 2 3 4

Réponse incorrecte

13.6

Réponse attendue

1 3 6

Question 14

Question à réponse unique

Quelle est la bonne définition d’un graphe eulérien ?

Réponses correctes

	Réponse attendue	Réponse saisie	Réponse discordante	
A	<input type="checkbox"/>	<input type="checkbox"/>	Non	Un graphe est eulérien s’il existe une chaîne qui passe par toutes les arêtes du graphe une et une seule fois.
B	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Non	Un graphe est eulérien s’il existe un cycle qui passe par toutes les arêtes du graphe une et une seule fois.
C	<input type="checkbox"/>	<input type="checkbox"/>	Non	Un graphe est eulérien s’il existe une chaîne qui passe par tous les sommets du graphe une et une seule fois.
D	<input type="checkbox"/>	<input type="checkbox"/>	Non	Un graphe est eulérien s’il existe un cycle qui passe par tous les sommets du graphe une et une seule fois.
E	<input type="checkbox"/>	<input type="checkbox"/>	Non	Un graphe est eulérien s’il existe un cycle qui passe par toutes les arêtes du graphe au moins une fois.

On considère le graphe représenté par la matrice d’adjacence suivante :

0	1	0	0	1
1	0	0	1	0
0	0	0	1	1
0	1	1	0	0
1	0	1	0	0

Quelles sont les affirmations vraies ?

Réponses correctes	0 discordance
--------------------	---------------

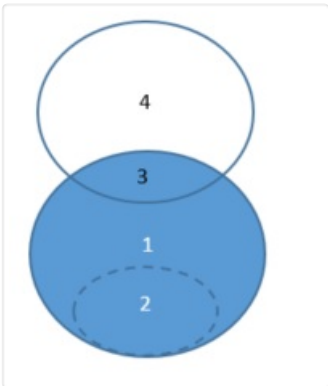
	Réponse attendue	Réponse saisie	Réponse discordante	
A	<input type="checkbox"/>	<input type="checkbox"/>	Non	Le graphe n’est pas connexe.
B	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Non	Le graphe est eulérien.
C	<input type="checkbox"/>	<input type="checkbox"/>	Non	Le graphe n’est pas hamiltonien.
D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Non	Le degré moyen du graphe est de 2.
E	<input type="checkbox"/>	<input type="checkbox"/>	Non	Le graphe est orienté.

Parmi ces problèmes, lesquels se modélisent bien avec des graphes ?

Réponses correctes	0 discordance
--------------------	---------------

	Réponse attendue	Réponse saisie	Réponse discordante	
A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Non	Effectuer le routage des données dans un réseau filaire
B	<input type="checkbox"/>	<input type="checkbox"/>	Non	Optimiser le placement d’objets dans des conteneurs
C	<input type="checkbox"/>	<input type="checkbox"/>	Non	Rechercher la présence d’un mot dans un texte
D	<input type="checkbox"/>	<input type="checkbox"/>	Non	Maximiser l’occupation de machines dans une usine
E	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Non	Établir un plan de table qui minimise le nombre de personnes assises à côté d’un ennemi

Placez les différentes catégories de classes de complexité aux différentes zones du schéma ci-dessous



Réponses correctes			0 discordance	
Élément à associer	Réponse attendue	Réponse saisie	Réponse discordante	
1	NP	NP	Non	
2	P	P	Non	
3	NP-complet	NP-complet	Non	
4	NP-difficile	NP-difficile	Non	

Associez les classes suivantes avec leurs définitions.

Réponses correctes		0 discordance	
Élément à associer	Réponse attendue	Réponse saisie	Réponse discordante
La classe P	La classe des problèmes de décision pouvant être résolus en temps polynomial	La classe des problèmes de décision pouvant être résolus en temps polynomial	Non
La classe NP	La classe des problèmes pouvant être résolue en temps polynomial par une machine de Turing non déterministe.	La classe des problèmes pouvant être résolue en temps polynomial par une machine de Turing non déterministe.	Non
La classe NP-difficile	La classe des problèmes au moins aussi difficile que tous les autres problèmes de la classe NP.	La classe des problèmes au moins aussi difficile que tous les autres problèmes de la classe NP.	Non
La classe NP-complet	La classe des problèmes pour lesquels il n'existe aucun algorithme polynomial qui peut le résoudre, mais dont une solution peut être vérifiée en temps polynomial.	La classe des problèmes pour lesquels il n'existe aucun algorithme polynomial qui peut le résoudre, mais dont une solution peut être vérifiée en temps polynomial.	Non

Le nombre d’opérations du programme suivant est :

```
1. def somme(n):
2.     s = 0
3.     for i in range(n+1):
4.         s += i
5.     return s
6.
7. print(somme(100))
```

Réponses incorrectes

	Réponse attendue	Réponse saisie	Réponse discordante	
A	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Oui (+1)	5n+1
B	<input type="checkbox"/>	<input type="checkbox"/>	Non	5n
C	<input type="checkbox"/>	<input type="checkbox"/>	Non	5n-1
D	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Oui (+1)	n
E	<input type="checkbox"/>	<input type="checkbox"/>	Non	n+1

La complexité de ce programme est :

```
1. def somme(n):
2.     s = 0
3.     for i in range(n+1):
4.         s += i
5.     return s
6.
7. print(somme(100))
```

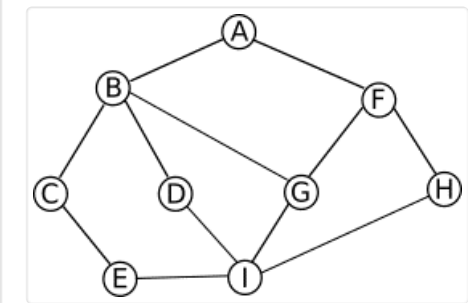
Réponses correctes

	Réponse attendue	Réponse saisie	Réponse discordante	
A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Non	Polynomiale
B	<input type="checkbox"/>	<input type="checkbox"/>	Non	Constante
C	<input type="checkbox"/>	<input type="checkbox"/>	Non	Logarithmique
D	<input type="checkbox"/>	<input type="checkbox"/>	Non	Quadratique
E	<input type="checkbox"/>	<input type="checkbox"/>	Non	Exponentielle

Si un algorithme a une complexité temporelle asymptotique $O(f(n))$:

Réponses correctes				
	Réponse attendue	Réponse saisie	Réponse discordante	
A	<input type="checkbox"/>	<input type="checkbox"/>	Non	Au pire il faut en ordre de grandeur n opérations à l'algorithme pour résoudre une instance de taille $f(n)$
B	<input type="checkbox"/>	<input type="checkbox"/>	Non	En moyenne il faut en ordre de grandeur $f(n)$ opérations à l'algorithme pour résoudre n'importe quelle instance.
C	<input type="checkbox"/>	<input type="checkbox"/>	Non	Au pire il faut en ordre de grandeur n opérations à l'algorithme pour résoudre une instance de taille n
D	<input type="checkbox"/>	<input type="checkbox"/>	Non	Au pire il faut en ordre de grandeur $f(n)$ opérations à l'algorithme pour résoudre une instance de taille $f(n)$
E	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Non	Au pire il faut en ordre de grandeur $f(n)$ opérations à l'algorithme pour résoudre une instance de taille n

Quel est l'ordre du graphe suivant ?



Réponses correctes		0 discordance	
	Réponse attendue	Réponse saisie	Réponse discordante
	[9 ; 9]	9	Non
Commentaire de correction de la proposition			

Mettez dans l'ordre de temps de calcul asymptotiquement croissant les ordres de grandeur de complexité algorithmique suivantes :

a : $O(n^2)$
b : $O(n)$
c : $O(2^n)$
d : $O(n^3)$
e : $O(n!)$
f : $O(\log(n))$

Réponses correctes

0 discordance

Élément à associer	Réponse attendue	Réponse saisie	Réponse discordante	
1	f	f	Non	
2	b	b	Non	
3	a	a	Non	
4	d	d	Non	
5	c	c	Non	
6	e	e	Non	

Que signifie $f(x) \in O(g(x))$?

Réponses correctes

0 discordance

	Réponse attendue	Réponse saisie	Réponse discordante	
A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Non	f est bornée par g à un facteur près pour un x suffisamment grand
B	<input type="checkbox"/>	<input type="checkbox"/>	Non	g est bornée par f à un facteur près pour un x suffisamment grand
C	<input type="checkbox"/>	<input type="checkbox"/>	Non	Il existe un algorithme de complexité f qui est bornée asymptotiquement par g
D	<input type="checkbox"/>	<input type="checkbox"/>	Non	Il existe un algorithme de complexité g qui est bornée asymptotiquement par f
E	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Non	Tout algorithme de complexité f peut être considéré comme un algorithme de complexité $O(g(x))$

On considère un algorithme calculant le degré maximal d'un graphe d'ordre n :

```
max = 0
Pour chaque sommet i du graphe :
  si max < degré(i)
    max = degré(i)
```

Dans une représentation par matrice d'adjacence, quelle est la complexité temporelle au pire de cet algorithme ?
L'opération degré(i) est réalisée en $O(1)$

Réponses correctes				
	Réponse attendue	Réponse saisie	Réponse discordante	
A	<input type="checkbox"/>	<input type="checkbox"/>	Non	constante
B	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Non	linéaire
C	<input type="checkbox"/>	<input type="checkbox"/>	Non	logarithmique
D	<input type="checkbox"/>	<input type="checkbox"/>	Non	quadratique
E	<input type="checkbox"/>	<input type="checkbox"/>	Non	cubique
F	<input type="checkbox"/>	<input type="checkbox"/>	Non	exponentielle

?

Question 26

Question à réponses multiples

On considère la complexité temporelle d'un algorithme.

Sélectionnez l(es) affirmation(s) exacte(s)

Réponses correctes

0 discordance

	Réponse attendue	Réponse saisie	Réponse discordante	
A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Non	Un ordinateur ne peut pas exécuter un algorithme de complexité exponentielle
B	<input type="checkbox"/>	<input type="checkbox"/>	Non	Une complexité polynomiale est moins bonne qu'une complexité linéaire
C	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Non	La complexité temporelle permet d'évaluer le temps que met un algorithme pour se terminer
D	<input type="checkbox"/>	<input type="checkbox"/>	Non	Un algorithme de complexité quadratique est inexploitable
E	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Non	La complexité temporelle est exprimée de manière asymptotique

Soit le problème de la coupe maximale dans un graphe

Données : Un graphe G

Problème : Quel est le nombre maximal d'arêtes que l'on peut retirer de G sans qu'il devienne non connexe ?

Le problème de décision correspondant est :

Réponses correctes

	Réponse attendue	Réponse saisie	Réponse discordante	
A	<input type="checkbox"/>	<input type="checkbox"/>	Non	Étant donné un graphe G , est-il possible de retirer k arêtes de G sans le déconnecter ?
B	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Non	Étant donnés un graphe G et un entier k , est-il possible de retirer k arêtes sans déconnecter G ?
C	<input type="checkbox"/>	<input type="checkbox"/>	Non	Étant donné un graphe G , existe-t-il un entier k tel qu'il est possible de retirer k arêtes à G sans le déconnecter ?
D	<input type="checkbox"/>	<input type="checkbox"/>	Non	Étant donné un graphe G , quel est l'entier k tel qu'il est possible de retirer k arêtes sans déconnecter G ?
E	<input type="checkbox"/>	<input type="checkbox"/>	Non	Étant donnés un graphe G et un entier k , quelles sont les k arêtes que l'on peut retirer de G sans le déconnecter ?