

## Quiz – Analyse de complexité des algorithmes

Nom : .....

Note :

Prénom : .....

..... /20

### Exercice 1.

*Arithmétique*

Dans cet exercice, on considère que la multiplication est une instruction élémentaire.

1. Voici un algorithme d'exponentiation simple.

Entrée : deux entiers positifs  $n$  et  $p$

Sortie :  $n^p$

```
res = 1
pour i de 1 à p faire:
    res = res * n
retourner res
```

Quelle est la complexité de cet algorithme?

- ☐  $\mathcal{O}(\log p)$     ☐  $\mathcal{O}(p)$     ☐  $\mathcal{O}(2^p)$   
☐ Logarithmique    ☐ Linéaire    ☐ Exponentiel

/1 point

/2 points

2. Voici un algorithme d'exponentiation rapide.

Entrée : deux entiers positifs  $n$  et  $p$

Sortie :  $n^p$

```
res = 1
tant que p > 0 faire:
    si p modulo 2 == 1 alors:
        res = res * n
        p = (p-1) / 2
    sinon:
        p = p / 2
    n = n * n
retourner res
```

Quelle est la complexité de cet algorithme?

- ☐  $\mathcal{O}(\log p)$     ☐  $\mathcal{O}(p)$     ☐  $\mathcal{O}(2^p)$   
☐ Logarithmique    ☐ Linéaire    ☐ Exponentiel

/1 point

/2 points

### Exercice 2.

*Arithmétique élémentaire*

On s'intéresse ici aux opérations élémentaires de l'arithmétique.

3. Quelle est la complexité de l'algorithme d'addition de deux entiers naturels que l'on apprend à l'école primaire?

- ☐ Logarithmique    ☐ Linéaire    ☐ Quadratique    ☐ Cubique

/1 point

4. Est-il possible qu'un algorithme théoriquement plus efficace existe?

- ☐ Oui    ☐ Non

/1 point

5. Quelle est la complexité de l'algorithme de multiplication de deux entiers naturels que l'on apprend à l'école primaire ?  
☐ Logarithmique    ☐ Linéaire    ☐ Quadratique    ☐ Cubique /1 point
6. Il existe un algorithme en  $\mathcal{O}(n \log n)$  pour multiplier deux entiers de  $n$  bits. En quelle année a-t-il été découvert ?  
☐ -300 av. J.-C.    ☐ 1845    ☐ 1968    ☐ 2019 /1 point

### Exercice 3.

Tris

On souhaite trier un tableau de  $n$  entiers compris entre 0 et 100.

7. Quelle est la complexité de l'algorithme du tri à bulles ?  
☐  $\mathcal{O}(n)$     ☐  $\mathcal{O}(n \log n)$     ☐  $\mathcal{O}(n^2)$     ☐  $\mathcal{O}(n^3)$  /1 point
8. Quelle est la complexité de l'algorithme du tri par insertion ?  
☐  $\mathcal{O}(n)$     ☐  $\mathcal{O}(n \log n)$     ☐  $\mathcal{O}(n^2)$     ☐  $\mathcal{O}(n^3)$  /1 point
9. Quelle est la complexité en moyenne de l'algorithme du tri rapide (Quicksort) ?  
☐  $\mathcal{O}(n)$     ☐  $\mathcal{O}(n \log n)$     ☐  $\mathcal{O}(n^2)$     ☐  $\mathcal{O}(n^3)$  /1 point
10. Quelle est la complexité (dans le pire cas) de l'algorithme du tri rapide (Quicksort) ?  
☐  $\mathcal{O}(n)$     ☐  $\mathcal{O}(n \log n)$     ☐  $\mathcal{O}(n^2)$     ☐  $\mathcal{O}(n^3)$  /1 point
11. Quelle est la complexité de la recherche dichotomique d'un élément dans un tableau trié ?  
☐  $\mathcal{O}(\log n)$     ☐  $\mathcal{O}(n)$     ☐  $\mathcal{O}(n^2)$     ☐  $\mathcal{O}(n \log n)$  /1 point

### Exercice 4.

Graphes

Pour un graphe  $G = (V, A)$ , on note  $n = |V|$  son nombre de sommets, et  $m = |A|$  son nombre d'arêtes (cas non-orienté) ou d'arcs (cas orienté).

12. Lesquelles de ces inégalités sont vraies pour tout graphe orienté  $G$  ?  
☐  $m \leq n$     ☐  $m \leq \frac{n(n+1)}{2}$     ☐  $m \leq n^2$     ☐  $m \leq n^3$     ☐  $n \leq m$  /1 point
13. Lesquelles de ces inégalités sont vraies pour tout graphe non-orienté  $G$  ?  
☐  $m \leq n$     ☐  $m \leq \frac{n(n+1)}{2}$     ☐  $m \leq n^2$     ☐  $m \leq n^3$     ☐  $n \leq m$  /1 point
14. Quelle est la complexité de calculer le degré  $\deg(v)$  d'un sommet  $v \in V$  dans un graphe non-orienté représenté par matrice d'adjacence ?  
☐  $\mathcal{O}(\log n)$     ☐  $\mathcal{O}(\deg(v))$     ☐  $\mathcal{O}(n)$     ☐  $\mathcal{O}(n^2)$     ☐  $\mathcal{O}(m)$  /1 point
15. Quelle est la complexité de rechercher si un graphe non-orienté donné possède un cycle eulérien (qui passe exactement une fois par chaque arête) ?  
☐  $\mathcal{O}(1)$     ☐  $\mathcal{O}(n)$     ☐  $\mathcal{O}(n + m)$     ☐  $\mathcal{O}(n^2)$     ☐  $\mathcal{O}(n^3)$  /1 point
16. Quelle est la complexité de l'algorithme de Dijkstra, qui permet de trouver un plus court chemin de  $s \in V$  à  $t \in V$  dans un graphe orienté dont les arcs sont pondérés par des nombres réels positifs ? (Cet algorithme calcule des plus courts chemin de  $s$  à tous les sommets du graphe  $G$ .) On suppose une implémentation simple avec deux tableaux.  
☐  $\mathcal{O}(m + n \log n)$     ☐  $\mathcal{O}(m^2 \log n)$     ☐  $\mathcal{O}(n^2)$     ☐  $\mathcal{O}(n m^2)$  /1 point