

Rapport d'évaluation des algorithmes



Table of Contents

Tests complémentaires effectués sur tous les algorithmes.....	2
Critères de classement pour le concours.....	3
Simplicité.....	3
Efficacité.....	4
Sobriété.....	4
Simplicité.....	5
Meilleur N°12.....	5
Meilleur N°38.....	6
Meilleur N°19.....	7
Pire N°36.....	8
Pire N°55.....	9
Efficacité.....	10
Meilleur N°2.....	10
Meilleur N°53.....	11
Pire N°17.....	12
Pire N°34.....	13
Sobriété.....	14
Meilleur N°4.....	14
Meilleur N°5.....	15
Pire N°34.....	16
Pire N°41.....	17

Simplicité / meilleur :	12, 38, 19
Simplicité / pire :	36, 55
Efficacité / meilleur :	2, 53
Efficacité / pire :	17, 34
Sobriété / meilleur :	4, 5
Sobriété / pire :	34, 41

Tests complémentaires effectués sur tous les algorithmes

Les tests N°2 et 4 auront un impact moins conséquent sur la note puisqu'il n'était pas explicitement demandé que la solution ne les passe.

N°	Test	Explication
1	<code>assertEquals(List.of("Orange", "Prune"), Exercice.solution("Prune Orange", List.of('O', 'P')));</code>	Vérifie si la solution parvient bien à trier deux mots commençant par des lettres différentes.
2	<code>assertEquals(List.of("orange", "Prune"), Exercice.solution("Prune orange", List.of('O', 'p')));</code>	Vérifie si la solution prend en compte les lettres minuscules et majuscules comme la même lettre
3	<code>assertEquals(List.of("0range", "Prune"), Exercice.solution("Prune 0range", List.of('0', 'P')));</code>	Vérifie si la solution peut prendre en compte les chiffres
4	<code>assertEquals(List.of("Ananas", "Citron", "Cerise", "Orange", "Prune", "Pomme", "Poire", "Gruyere"), Exercice.solution("Orange Poire Ananas Gruyere Citron Cerise ~# Prune Pomme", List.of('A', 'C', 'R', 'O', 'P')));</code>	Tests supplémentaires pour vérifier que le programme trie bien les mots en fonction de toutes leurs lettres.

Critères de classement pour le concours

Pour chaque catégorie, tous les algorithmes seront étudiés en fonction des éléments de notation suivants. Pour les catégories "Meilleur", ils seront classés sur le podium du plus haut au plus bas en fonction de leur note. Pour les catégories "Pire", ils seront étudiés de la même façon, mais seront classés dans l'ordre inverse (La moins bonne note ira le plus haut sur le podium). Le tableau ci-dessous sera valable pour toutes les catégories, pour la catégorie "Pire", elle enlève des points au lieu d'en ajouter.

Condition	Points
Passe les tests de base de la grille d'évaluation	/20
Passe les tests complémentaires	/12

Simplicité

Conditions	Points
Les variables sont clairement nommées	/5
Le code est commenté lorsque nécessaire	/3
Bonne mise en forme globale (indentation, etc...) Code bien ordonné et généralement bien lisible	/5

Efficacité

Conditions	Points
<p>Vitesse d'exécution du code</p> <pre>long startTime = System.nanoTime();</pre> <pre>long endTime = System.nanoTime();</pre> <pre>long executionTime = endTime - startTime;</pre> <p>Le jeu d'essai est</p> <ul style="list-style-type: none">une liste de 10 mots et 4 lettresune liste de 100 mots et 4 lettresune liste de 1000 mots et 4 lettres	

Sobriété

Conditions	Points
le nombre de variables	/5
Type de variable	/2
Taille du fichier	Tri par taille de fichiers. Le plus petit gagne + 1 le suivant gagne + 0.5 le troisième gagne + 0.25

Simplicité

Meilleur N°12

Le code compile, respecte l'anonymat et la nomenclature précise. Il ne passe pas tous les tests cependant.

Tests de base

Condition	Points
Passe les tests de base de la grille d'évaluation	15/20
Passe les tests complémentaires	6/12

Simplicité

Conditions	Points
Les variables sont clairement nommées	5/5
Le code est commenté lorsque nécessaire	1/3
Bonne mise en forme globale (indentation, etc...) Code bien ordonné et généralement bien lisible	5/5

Total : 32

Meilleur N°38

Le code compile, respecte l'anonymat et la nomenclature précise. Il ne passe aucun test.

Tests de base

Condition	Points
Passe les tests de base de la grille d'évaluation	0/20
Passe les tests complémentaires	0/12

Simplicité

Conditions	Points
Les variables sont clairement nommées	5/5
Le code est commenté lorsque nécessaire	3/3
Bonne mise en forme globale (indentation, etc...) Code bien ordonné et généralement bien lisible	5/5

Total : 13

Meilleur N°19

Le code compile, respecte l'anonymat. Il ne respecte pas la nomenclature précise. Il passe tous les tests initiaux ainsi que 2/4 tests complémentaires. Présence d'une javadoc.

Tests de base

Condition	Points
Passe les tests de base de la grille d'évaluation	20/20
Passe les tests complémentaires	6/12

Simplicité

Conditions	Points
Les variables sont clairement nommées	5/5
Le code est commenté lorsque nécessaire	3/3
Bonne mise en forme globale (indentation, etc...) Code bien ordonné et généralement bien lisible	5/5

Total : 39

Pire N°36

Le code compile, respecte l'anonymat. Il ne respecte pas la nomenclature précise. Il passe tous les tests initiaux ainsi que 2/4 tests complémentaires. Les variables consistent en séries de 1 lettre et 1 chiffre, ce qui les rend très peu lisibles. Le code n'est pas commenté, la mise en forme n'aide pas à la lecture.

Tests de base

Condition	Points
Passe les tests de base de la grille d'évaluation	-20/-20
Passe les tests complémentaires	-6/-12

Simplicité

Conditions	Points
Les variables sont clairement nommées	-5/-5
Le code est commenté lorsque nécessaire	-3/-3
Bonne mise en forme globale (indentation, etc...) Code bien ordonné et généralement bien lisible	-5/-5

Total : -39

Pire N°55

Le code compile, respecte l'anonymat. Il respecte la nomenclature précise. Il passe tous les tests initiaux ainsi que 2/4 tests complémentaires. Les variables sont toutes nommées d'1 lettre, ce qui les rend très peu lisibles. Le code est commenté de commentaires inutiles, la mise en forme n'aide pas à la lecture.

Tests de base

Condition	Points
Passe les tests de base de la grille d'évaluation	-10/-20
Passe les tests complémentaires	-3/-12

Simplicité

Conditions	Points
Les variables sont clairement nommées	-5/-5
Le code est commenté lorsque nécessaire	-3/-3
Bonne mise en forme globale (indentation, etc...) Code bien ordonné et généralement bien lisible	-5/-5

Total : -26

Efficacité

Meilleur N°2

Le code compile, respecte l'anonymat et la nomenclature précise. Il ne passe pas tous les tests cependant. Aucun test initial ne passe, et deux tests complémentaires passent sur 4. Le test rédigé en C ne fonctionne sûrement pas comme prévu, ainsi la notation n'est pas trop sévère.

Efficacité

Conditions	Points
Pas réussi à effectuer les tests	

Total : /

Meilleur N°53

Le code compile, respecte l'anonymat. Il ne respecte pas la nomenclature précise.

Efficacité

Conditions	Points
Jeu d'essai de 10 : 4 Jeu d'essai de 100 : 20 Jeu d'essai de 100 : 184	208

Total : 208

Pire N°17

Le code compile, respecte l'anonymat. Il ne respecte pas la nomenclature précise.

Efficacité

Conditions	Points
Jeu d'essai de 10 : 49 Jeu d'essai de 100 : 67 Jeu d'essai de 100 : 203	319

Total : 319

Pire N°34

Le code compile, respecte l'anonymat. Il ne respecte pas la nomenclature précise.

Efficacité

Conditions	Points
Jeu d'essai de 10 : 44 Jeu d'essai de 100 : 17 Jeu d'essai de 100 : 49	110

Total : 110

Sobriété

Meilleur N°4

Tests de base

Condition	Points
Passe les tests de base de la grille d'évaluation	20/20
Passe les tests complémentaires	6/12

Sobriété

Conditions	Points
le nombre de variables	5/5
Type de variable	2/2
Taille du fichier	889 octets (889 octets)

Total : 33

Meilleur N°5

Tests de base

Condition	Points
Passe les tests de base de la grille d'évaluation	10/20
Passe les tests complémentaires	6/12

Sobriété

Conditions	Points
le nombre de variables	5/5
Type de variable	2/2
Taille du fichier	793 octets (793 octets)

Total : 23

Pire N°34

Tests de base

Condition	Points
Passe les tests de base de la grille d'évaluation	15/20
Passe les tests complémentaires	6/12

Sobriété

Conditions	Points
le nombre de variables	5/5
Type de variable	2/2
Taille du fichier	3,2 Ko (3229 octets)

Total : 28

Pire N°41

Tests de base

Condition	Points
Passe les tests de base de la grille d'évaluation	10/20
Passe les tests complémentaires	3/12

Sobriété

Conditions	Points
le nombre de variables	5/5
Type de variable	2/2
Taille du fichier	5,1 Ko (5062 octets)

Total : 20

