Devoir de Programmation : Tries

Marco PONTI
Nicolas SERRES

December 2016

1 Introduction

Question 1.1 : Le caractre pour le fin d'un mot est '@'

Question 4.10 : Calcule des complexites PatriciaTrie :

findPrefix = O (longeur du mot)
(Parcours d'une chaine de caractre et comparaisons pour chaque caractre)

displayPtree = O (nb max de caractre * profondeur)
(Parcours d'une Hashmap au pire des cas contenant tout les caratres possible)

cloneAll = O (nb max de caractre)
(Parcours d'une Hashmap au pire des cas contenant tout les caratres possible)

search = O (4 * (longeur du mot))
(Une comparaison puis lancement de la sous fonction 4 comparaisons pour chaque caractre du mot aux pire cas)

delete = O (4 * (longeur du mot))
(Une comparaison puis lancement de la sous fonction 4 comparaisons pour chaque caractre du mot aux pire cas)

insert = O ((5 * cloneAll) * longeur du mot)
(5 comparaison puis lancement de la fonction cloneAll, au pire cas pour chaque caractres du mot)

countWord = O (nb max de caractre * profondeur)
(Une comparaisons et parcours d'une Hashmap au pire des cas contenant tout les caratres possible pour toute la profondeur de l'arbre)

countDeep = O (nb max de caractre * profondeur)
(Une comparaisons et parcours d'une Hashmap au pire des cas contenant tout

les caratres possible pour toute la profondeur de l'arbre)

arrayWord = O (nb max de caractre * profondeur)
(Une comparaisons et parcours d'une Hashmap au pire des cas contenant tout les caratres possible pour toute la profondeur de l'arbre)

allWord = O (nb max de caractre * profondeur)
(Une comparaisons et parcours d'une Hashmap au pire des cas contenant tout les caratres possible pour toute la profondeur de l'arbre)

copy = O (nb max de caractre * profondeur)
(Parcours d'une Hashmap au pire des cas contenant tout les caratres possible pour toute la profondeur de l'arbre)

split = O (cloneAll)
(Deux comparaisons et lancement de la fonction cloneAll)

fusion = O (2 * (nb max de caractre * profondeur-min-d'un-des-deux-arbres))
(Deux comparaisons puis, parcours d'une Hashmap au pire des cas contenant tout les caratres possible pour toute la profondeur de l'arbre le plus court)

getDeep = O (nb max de caractre * profondeur)
(Une comparaisons et parcours d'une Hashmap au pire des cas contenant tout les caratres possible pour toute la profondeur de l'arbre le plus court)

mediumDeep = O (getDeep * profondeur)
(Appel getDeep puis Parcours la liste obtenu)

getPrefix = O (3 * longeur du mot)
(Trois comparaisons pour chaque caractre du mot)

convert = O ((nb caractre du prfix + nb max de caractre) * profondeur)
(Parcours du prfixe et parcours de la Hashmap des fils au pire des cas contenant tout les caratres possible pour toute la profondeur de l'arbre)

Calcule des complexites au pire Tries Hybrides : (nombre de comparaison)

ajouterMot = O (5 * (longeur du mot))

recherche = O (4 * (longeur du mot))

comptageMots = O (2 * nb de noeud)

listeMots = O 1 + (4 * nb de noeud))

comptageNil = O (nb de noeud)

hauteur = O (nb de noeud)

profondeurMoyenne = O (4 * nb de noeud)

prefixe = O ((4 * longeur du prefixe) + (2 * nb de noeud))

suppression = O (4 * (longeur du mot)) + (hauteur de l'arbre) + (4 * longeur du mot * hauteur de l'arbre)

Conversion Hybrides vers Patricia = O ((4 * nb de noeud) + (nb de mot * cloneAll * (longeur du mot)))

Question 5.11 et 5.12 : Benchmark)

Tries Hybrides Benchmark (en nanoseconde)

file	build	insert	search	delete	nbword	deep
lll.txt	66631126	5400	32479	9944	3634	33
comedy_errors.txt	35247477	4362	8450	9906	2435	24
richardii.txt	58213606	12322	24619	19888	3510	26
twelfth_night.txt	39637143	6437	11022	23118	3027	31
merchant.txt	40999826	4720	36604	3085	3157	28
merry_wives.txt	40514232	4609	2418	4143	3185	27
henryv.txt	51744248	5229	2616	6405	4390	30
1henryiv.txt	72914001	3274	2324	3931	3720	27
2henryiv.txt	39277108	24658	3824	5499	3960	32
1henryvi.txt	45192574	23767	2051	4798	3722	25
3henryvi.txt	46620362	9555	4040	10040	3445	29
measure.txt	38081410	4580	2601	3625	3226	24
othello.txt	49834188	7369	3361	6280	3662	26
taming_shrew.txt	60186504	6991	2155	6883	3146	25
tempest.txt	50169038	8435	8249	6609	3081	27
macbeth.txt	56239471	5156	3291	4477	3201	26
asyoulikeit.txt	47967582	4779	2463	5888	3167	28
allswell.txt	50410989	7507	2326	5551	3385	27
hamlet.txt	66066594	6046	2862	5910	4551	25
julius_caesar.txt	24679242	21777	2212	3582	2790	26
much_ado.txt	18830476	17983	2026	1869	2904	26
coriolanus.txt	21227378	4439	3173	6927	3880	25
romeo_juliet.txt	22072252	3407	2199	4169	3542	26
timon.txt	14088450	4331	3013	4623	3184	26
winters_tale.txt	20839806	3880	2483	5687	3711	28
pericles.txt	17679759	6127	3612	4068	3141	24
john.txt	38864785	7667	3433	4169	3434	27
2henryvi.txt	78783272	6920	8614	6489	3932	32
richardiii.txt	49770234	4147	2623	4113	3893	25
henryviii.txt	27955676	4286	3619	5851	3512	26
troilus_cressida.txt	32187681	4799	2707	5803	4116	27
midsummer.txt	19945487	6235	3323	7147	2911	25
cymbeline.txt	24953860	10306	6786	12717	4054	26
lear.txt	21814093	3153	2240	3296	4004	26
cleopatra.txt	22975703	2505	1963	4496	3779	28
titus.txt	19535893	1598	2284	2449	3303	26
two_gentlemen.txt	12821391	1540	2213	2580	2635	25
Moyen	39053322	7305	5845	6379	3468	26
Total	1444972917	270309	216290	236028	128329	994

Tries Hybrides Benchmark (en nanoseconde)

file	build	insert	search	delete	nbword	deep
lll.txt	66631126	5400	32479	9944	3634	33
comedy_errors.txt	35247477	4362	8450	9906	2435	24
richardii.txt	58213606	12322	24619	19888	3510	26
twelfth_night.txt	39637143	6437	11022	23118	3027	31
merchant.txt	40999826	4720	36604	3085	3157	28
merry_wives.txt	40514232	4609	2418	4143	3185	27
henryv.txt	51744248	5229	2616	6405	4390	30
1henryiv.txt	72914001	3274	2324	3931	3720	27
2henryiv.txt	39277108	24658	3824	5499	3960	32
1henryvi.txt	45192574	23767	2051	4798	3722	25
3henryvi.txt	46620362	9555	4040	10040	3445	29
measure.txt	38081410	4580	2601	3625	3226	24
othello.txt	49834188	7369	3361	6280	3662	26
taming_shrew.txt	60186504	6991	2155	6883	3146	25
tempest.txt	50169038	8435	8249	6609	3081	27
macbeth.txt	56239471	5156	3291	4477	3201	26
asyoulikeit.txt	47967582	4779	2463	5888	3167	28
allswell.txt	50410989	7507	2326	5551	3385	27
hamlet.txt	66066594	6046	2862	5910	4551	25
julius_caesar.txt	24679242	21777	2212	3582	2790	26
much_ado.txt	18830476	17983	2026	1869	2904	26
coriolanus.txt	21227378	4439	3173	6927	3880	25
romeo_juliet.txt	22072252	3407	2199	4169	3542	26
timon.txt	14088450	4331	3013	4623	3184	26
winters_tale.txt	20839806	3880	2483	5687	3711	28
pericles.txt	17679759	6127	3612	4068	3141	24
john.txt	38864785	7667	3433	4169	3434	27
2henryvi.txt	78783272	6920	8614	6489	3932	32
richardiii.txt	49770234	4147	2623	4113	3893	25
henryviii.txt	27955676	4286	3619	5851	3512	26
troilus_cressida.txt	32187681	4799	2707	5803	4116	27
midsummer.txt	19945487	6235	3323	7147	2911	25
cymbeline.txt	24953860	10306	6786	12717	4054	26
lear.txt	21814093	3153	2240	3296	4004	26
cleopatra.txt	22975703	2505	1963	4496	3779	28
titus.txt	19535893	1598	2284	2449	3303	26
two_gentlemen.txt	12821391	1540	2213	2580	2635	25
Moyen	39053322	7305	5845	6379	3468	26
Total	1444972917	270309	216290	236028	128329	994

Conclusion : Sur quelque instance le Patricia-Tries est meilleur que le Hybride-Tries, nanmoins, en moyen l'Hybride est moins performent en construction mais ensuite il est plus rapide que le Patricia-Tries pour les fonctions d'insertion, d'ajout, de suppression et de recherche