Université Pierre et Marie Curie

Modelisation Objet

•

TME3 – Conteneurs & Itérateurs

Auteur:
William Fabre

Professeur:
Madame Marie-Minerve Louerat

Année 2018-2019

Contents

Question 1 : Écrire une fonction backInsert()

1.1 Charger dans un vecteur de string le texte en insérant les nouveaux éléments à la fin.

Pour se faire il faut declarer un vector¡std::string¿ et parcourir notre string qui est dans GPL_2_text.h. On va ajouter tous les mots avec la fonction push back des vectors.

1.2 Afficher le nombre d'éléments du vecteur.

pour afficher tous les elements j'ai declarer une fonction sous forme de template pour la reutiliser dans le maximum de cas possibles

```
template<typename T>
void show(T gpl);
```

Cette fonction affiche la size(returns the number of elements) et la max size(returns the maximum possible number of elements) puis affiche sur une seule ligne toutes les string contenue dans la structure contenant les strings. Cette fonction peut donc marcher pour les vectors et les lists mais pas les map. J'utilise aussi la fonction capacity(returns the number of elements that can be held in currently allocated storage)

1.3 Trier les éléments du vecteur.

Pour trier tous les elements du vecteur on va utiliser le template std::sort. Les elements sont compare en utilisant operator¡ deja implemente pour les vectors, lists et map. On utilise donc pas d'operateur de comparaison particulier et le protocol reste sequentiel car on ne fait pas de parallelisation. On precise donc juste le begin et le end.

1.4 Afficher tous les éléments du vecteur. On les affichera sur une seule ligne (ce sera très long).

la fonction show fait l'afficache en parcourant la structure de donnee contenant directement des strings grace a un iterateur sous forme de foreach. Cela fonctionnera pour les lists et les vectors.

1.5 Calcule de temps

le temps pour back_insert() est :

```
backInsert tt: 0.5 secondes ecoulees
```

La structure est faite sous forme de tableau et va donc inserer en fin de vector a la premiere place libre. Il possede deja un pointeur sur la premiere case libre et n'a donc pas besoin de parcours. L'insertion est en O(1) et on a donc un temps assez court modulo l'affichage et le remplissage complet du vector avec les doublons.

Question 2 : Écrire une fonction frontInsert() identique à la précédente

Pour se faire on va utiliser:

```
iterator insert( iterator pos, const T& value );
```

Cette fonction est utilisable par les vectors et va permettre de preciser qu'on doit inserer dans vector.begin().

le temps est de :

```
frontInsert tt: 0.94 secondes ecoulees
```

plus long car on est oblige de revenir au debut a chaque fois et il est oblige de deplacer tous les elements suivant dans le vecteurs ce qui n'est pas efficace.

Question 3: Écrire une fonction sortEachInsert()

Le temps est de :

```
backInsert && Sort each tt: 3.656 secondes ecoulees
```

On est ici sur un temps beaucoup plus long car le rangement du vecteur va se faire a chaque insertion est il faut pour ranger un vecteur au mieux O(nlog(n). cf reference : "O(N·log(N)), where N = std::distance(first, last) comparisons on average. (until C++11)"

Question 4 : Le conteneur List

Voici l'affichage dans le terminal pour les differentes fonctions.

```
backInsert tt: 0.4 secondes ecoulees
frontInsert tt: 0.4 secondes ecoulees
frontInsert && sort each tt: 2.308 secondes ecoulees
```

Le back insert ne pose pas de probleme dans une liste doublement chainee car on possede un iterateur dans les deux sens de la liste. On peut donc inserer en tete et en queue en O(1). Avec les fonctions : push_front (inserts an element to the beginning) et push_back (adds an element to the end). Ici on utilise push et pas emplace car ce sont des elements simple et on cherche juste a faire une copie d'un element dans notre vecteur. cf : "if you want to add a copy of an existing instance of the class to the container, use push. If you want to create a new instance of the class, from scratch, use emplace." https://stackoverflow.com/questions/26198350/c-stacks-push-vs-emplace/26198609

Question 5 : Le conteneur map

5.1 Chercher si un élément ayant pour clé le mot existe.

Pour savoir si un mot existe on va utiliser la fonction finds et non pas la fonction contains. Elles possedent apparemment la meme complexite algorithmique mais la fonction finds va nous renvoyer un iterateur s'il trouve un element correspondant a la clee recherchee. On va pouvoir verifier si cet iterateur est juste apres le dernier element car finds va fouiller tout jusqu'au dernier element. La complexite est Logarithmiquepar rapport a la taille de la map. Si on est apres la fin alors on n'a rien trouve sinon on a trouve un element donc on incremente le champ valeur. Dans l'autre cas on ajoute un nouvel element avec la fonction insert. en declarant un objet pair.

5.2 Afficher le nombre d'éléments

De maniere plus complexe on peut creer une structure qui tient a jour le nombre d'element mais je me suis contente de faire un parcours de tous les elements avec une variable accumulatrice.

```
Total number of words :2981
```

5.3 Afficher l'ensemble des éléments de la map

Map TB

0 02110 1 10 11 12 1301 1989 1991 2 3 4 5 51 6 69 7 8 9 A ABOVE ABSOLUTELY ADVISED AGREED ALL AND ANY APPLICABLE ARISING AS ASSUME Accompany Activities Also And Any Apply April BE BECAUSE BEEN BEING BUT BY Boston But By C CHARGE CONDITIONS CONSEQUENTIAL COPYING COPYRIGHT CORRECTION COST Coon Copyright DAMAGES DATA DEFECTIVE DISTRIBUTION EITHER END ENTIRE EVEN EVENT EXCEPT EXPRESSED EXTENT Each Everyone Exception FAILURE FITNESS FOR FREE Fifth Finally Floor For Foundation Foundations Franklin Free GENERAL GNU General Gnomovision HAS HOLDER HOLDERS Hacker Here Hereinafter How However IF IMPLIED IN INABILITY INACCURATE INCIDENTAL INCLUDING IS ISWITHOUT If In Inc It James June KIND LAW LIABLE LICENSE LICENSED LIMITED LOSS LOSSES Lesser License Licenses MA MAY MERCHANTABILITY MODIFICATION MODIFY Many NECESSARY NO NOT New OF OPERATE OR OTHER OTHERWISE OUT Of Our PARTICULAR PARTIES PARTY PERFORMANCE PERMITTED POSSIBILITY PROGRAM PROGRAMS PROVE PROVIDE PUBLIC PURPOSE Preamble President Program Programs Public QUALITY REDISTRIBUTE RENDERED REPAIR REQUIRED RISK SERVICING SHOULD SPECIAL STATED SUCH SUSTAINED Section Sections See Software Some Street

Subsection Such TERMS THE THERE THIRD TO Terms The Therefore These This Thus To Ty UNLESS USA USE Version Vice WARRANTIES WARRANTY WHEN WHO WILL WITH WITHOUT WRITING We When Whether YOU You Your Yoyodyne a above absence accept acceptance access accompanies accord achieve act actions add addition address addressed aggregation agreement all allegation allowed along also alter alternative among an and announcement another any anyone anything application applications applies apply appropriate appropriately are as ask associated at attach attempt author authors automatically avoid away b balance based be been believed below best binary body both brief bring but by c called can cannot carry case cause certain change changed changing charge choice choose circumstance circumstances claim claims clear clicks code collective comes commands commit compelled compilation compiler compilers complete compliance component components concerns conditions consequence consider considered consistent conspicuously constantly constitute contact containing contains contents contest contradict contrast contributions control convey copies copy copying copyright copyrighted copyrightline corresponding cost could countries counts course court covered customarily danger date decide decision definition deny depends derivative derivatives derived designated designed detail details develop differ different directly disclaimer disclaimerfor disclaims display distinguishing distribute distributed distributing distribution do document does donor each effect effectively either electronic else employer enforcing entire entirely equivalent even ever every everyone everyones example except exception exceptions exchange excluded excluding exclusion excuse executable exercise explicit expressly extend fee file files follow following for forbid form forming found free freedom from full further generally generous geographical get give given gives goals granted grants gratis greatest guarantee guided has have having he held hereby herein holder hope how hypothetical idea identifiable if implemented implied impose imposed in include included including incorporate incorporates incorporating independent indicate indirectly individually induce information infringe infringement installation instead intact integrity intended intent interactive interactively interchange interest interface interfaces into introduced invalid is issues it items its itself judgment keep kernel know language later law least legal library license licensed licensee licensees licenses licensor like limitation limited line linking long machine made mail major make makes making may means medium meet menu mere mode modification modifications modified modify modifying modules more most mouse must name names necessary need new no noncommercial normally not nothing notice notices number object obligations obtain of offer offering on one only operating option or order ordinary original other others otherwise our ours output outside paper part particular parties parts party passed passes patent patents people performing permission permissions permit permitted pertinent physical physically pieces place placed places plus pointer portion possible practices precise preferred present preserving prevent price print problems program programmer programs prohibited prominent promoting property proprietary protect protecting protection provide provided public publish published purpose range rather readable reads reason reasonably receive received receives recipient recipients redistribute redistribution redistributors refer referring refers reflect refrain regardless reliance remain reputations required requirements responsibilities responsible rest restricted restrictions reuse revised right rights royalty run running runs safest same sample satisfy say saying school scope scripts section sections separate service share sharing she short should show sign signature signed similar simultaneously since so software sole someone something sometimes source speak special specifies specify spirit start started starts stating status steps storage subject sublicense subroutine such suits sure surrender system take telling term terminate terminated terms than that the their them themselves then there thereof these they things third this thoroughly those though threatened three through thus time to too transferring translate translated translation true two type under understands unenforceable unless up use used useful user users using valid validity verbatim version versions view void volume w want warranty way we welcome what whatever when where whether which who whole whose wide will willing wish with without work works would write written wrote year years you your

map test : 0.4 secondes ecoulees

Question 6 : Fonction de Tri, Objet Fonctionnel

Pour creer cette fonction on va declarer une struct dans laquelle on va faire un overload sur la fonction operator(). Je reverse les deux strings grace a la fonction reverse avec le begin et le end de chaque string. On peut desormais renvoyer le lexicographic_compare entre les deux strings en prenant le begin et le end des deux strings inversees. On voit bien que les strings sont rangees dans l'ordre inverse des lettres.

On tombe sur ce resulstat :

Map TB

Total number of words :2981

0 10 02110 1 1301 11 51 1991 2 12 3 4 5 6 7 8 9 69 1989 A MA USA DATA C PUBLIC AGREED IMPLIED SUSTAINED RENDERED REQUIRED ADVISED LICENSED EXPRESSED STATED LIMITED PERMITTED SHOULD AND END KIND THIRD BE PERFORMANCE PROVIDE FREE CHARGE THE APPLICABLE LIABLE ASSUME THERE ENTIRE FAILURE OTHERWISE LICENSE PURPOSE USE BECAUSE OPERATE INACCURATE REDISTRIBUTE DEFECTIVE ABOVE PROVE IF OF SERVICING INCLUDING BEING ARISING WRITING COPYING SUCH WITH RISK SPECIAL CONSEQUENTIAL GENERAL INCIDENTAL ALL WILL PROGRAM BEEN WHEN EVEN IN MODIFICATION CORRECTION DISTRIBUTION WHO NO TO PARTICULAR HOLDER EITHER OTHER REPAIR OR FOR AS HAS DAMAGES WARRANTIES PARTIES LOSSES IS PROGRAMS TERMS CONDITIONS HOLDERS UNLESS FITNESS LOSS COPYRIGHT EXTENT EVENT NOT EXCEPT COST BUT OUT WITHOUT ISWITHOUT GNU YOU LAW MAY BY MODIFY ABSOLUTELY ANY NECESSARY QUALITY INABILITY MERCHANTABILITY POSSIBILITY WARRANTY PARTY a idea b c Public public electronic Inc instead add placed introduced guided provided intended included excluded need changed published modified implied called compelled threatened signed designed considered covered required preferred based revised licensed imposed passed addressed used associated translated designated terminated restricted copyrighted prohibited limited granted implemented started permitted distributed believed received derived allowed forbid valid invalid void avoid held could should would And and extend found third accord We be interface place Vice choice price notice service reliance compliance balance acceptance circumstance absence consequence since source induce made decide outside provide wide code mode include See fee Free free three licensee guarantee storage language change interchange exchange range infringe charge he The she the make take like readable unenforceable identifiable executable responsible possible Preamble file whole sole sample example people name same time Some welcome volume machine line copyrightline subroutine one someone anyone Everyone everyone June Yoyodyne scope hope type are share Software software Here there where mere entire Therefore more sure signature case These these precise exercise otherwise else License license sublicense those whose choose impose purpose course use cause excuse reuse mouse indicate date appropriate translate terminate separate incorporate complete write wrote distribute redistribute constitute true have achieve receive give alternative derivative interactive collective above we If Of brief if itself of thereof enforcing corresponding

including excluding changing distinguishing something nothing anything making linking telling willing forming performing containing running sharing bring offering referring transferring using operating incorporating stating protecting promoting distributing having preserving following saying modifying copying long along among Each each attach which Such such though through publish wish Fifth with both speak work ask geographical physical hypothetical legal special noncommercial original General kernel mail detail April all will full school control useful Program program them system claim verbatim freedom from term form medium In an can than been When then when written even given sign in remain refrain obtain certain herein Franklin on decision Gnomovision Version version permission exclusion modification application Foundation allegation aggregation compilation installation translation information limitation Section section Subsection protection addition definition Exception exception option portion distribution redistribution Coon reason Boston run To do who no too so Also also to into two keep develop up clear year similar particular number consider holder surrender under order refer differ offer danger rather Whether whether either other another further Hacker compiler disclaimer programmer paper Lesser user later Hereinafter alter pointer ever whatever However employer their or For for disclaimerfor author major donor Floor licensor Our our Your your as has reads commands understands depends interfaces places pieces practices notices circumstances licensees specifies applies accompanies copies countries responsibilities Activities parties makes files modules James names sometimes comes everyones does Licenses licenses passes incorporates issues receives gives derivatives themselves things is This this gratis clicks works goals details Programs programs problems items claims disclaims Terms terms means contains versions permissions modifications applications Foundations obligations reputations actions Sections sections restrictions conditions exceptions contributions concerns runs steps years refers others compilers users authors redistributors ours access regardless unless address rights its suits grants recipients requirements components patents contents counts scripts parts starts Thus thus plus generous status It at that what act intact contact effect object subject reflect protect contradict meet Street get right Copyright copyright it explicit commit permit spirit want President independent recipient equivalent announcement agreement infringement judgment document prominent pertinent component different present patent intent consistent prevent print not cannot accept except attempt part start short court least contrast best safest rest interest greatest contest cost most must But but without output menu You you w law New view new How how show below follow know By Ty display may say way away by hereby body they convey specify modify satisfy reasonably entirely appropriately interactively effectively thoroughly customarily physically automatically normally Finally generally individually only Apply apply expressly simultaneously conspicuously directly indirectly constantly Any any Many Accompany deny copy binary ordinary library necessary proprietary every carry validity integrity royalty warranty party property