

Test Trie

Objecte prova

És un test unitari per testejar si la classe Trie funciona correctament.

Aquesta és una classe útil quan volem facilitar la cerca d'autors del gestor de documents que contenen el prefix especificat. Aquest test comprova que l'estructura de dades en forma d'arbre fa correctament les funcions d'afegir, eliminar, comprovar si un string es troba a l'estructura i retornar totes les paraules d'aquesta que contenen el prefix indicat.

Valors estudiats

Totes les proves realitzades en aquest test són de caixa blanca. És a dir, la persona que ha programat la classe és la mateixa que la que l'ha testejat. Per tant, el programador coneix l'estructura interna del codi i sap tots els casos possibles que existeixen.

En aquest cas, la classe Trie és una classe molt simple formada per un arbre al qual podem afegir, eliminar i cercar prefixos.

- `retornaAutors()`

L'objectiu d'aquest test és comprovar que s'afegeixen correctament strings al trie. Per fer-ho, també comprovarem que ens retorna correctament els autors de l'estructura a partir d'un prefix.

Primerament crearem un `ArrayList<String>` on guardarem el resultat que esperem obtenir després d'afegir els autors. Tots els autors que hem afegit comparteixen el mateix prefix "joa". A continuació, crearem un trie i hi afegirem els mateixos autors que a l'`ArrayList`. Llistarem tots els autors amb prefix "joa" i els guardarem a un altre `ArrayList<String>` amb el resultat obtingut. Comprovarem que als dos `ArrayList` hi ha els mateixos autors.

En segon lloc, estudiarem el cas en el que demanem llistar tots els autors amb un prefix que no existeix al trie. Per tant, l'`ArrayList` amb el resultat que esperem obtenir serà buit. Creem un nou trie i hi afegim un autor amb nom "marc". Creem un nou `ArrayList<String>` i demanem que se'ns retorni tots els autors del nou trie amb prefix "joa". Veurem que la llista d'autors que es retorna en aquest cas també es buida.

En tercer lloc, creem un nou trie i hi afegim autors. Eliminem un d'ells i llistem tots els autors del trie (per fer-ho demanem que es llistin tots els autors amb prefix buit ""). Comprovem que la funció només retorna els autors que no han sigut eliminats.

- `removeautor()`

L'objectiu d'aquest test és comprovar s'eliminen correctament els strings indicats.

En primer lloc, creem un `ArrayList<String>` on guardarem el resultat que esperem obtenir després d'afegir i eliminar els autors. A continuació, afegim i eliminem els autors, i observem que efectivament els autors que ens retorna la funció quan demanem que ens els llisti tots (fent que el prefix sigui "") són tots els que hem afegit i no hem eliminat.

En segon lloc, comprovem que si eliminem una paraula que existeix al trie i alhora es prefix d'una altra, aquesta segona no s'elimina. És a dir, si afegim al trie les paraules "pal" i "palau" i eliminem "pal", comprovem que es llista "palau" ja que aquesta no l'hem eliminat. Per fer-ho creem un trie i al qual afegim les dues paraules mencionades anteriorment, i eliminem la que és prefix. Veiem que el resultat obtingut és efectivament només "palau".

En darrer lloc, comprovem que si eliminem una paraula que existeix al trie i que conté una altra paraula existent també al trie, aquesta segona no s'elimina. És a dir, si afegim al trie les paraules "pal" i "palau" i eliminem "palau", comprovem que es llista "pal" ja que aquesta no l'hem eliminat. Per fer-ho creem un trie al qual afegim les dues paraules mencionades anteriorment, i eliminem la que no es prefix. Veiem que el resultat obtingut és efectivament només "pal".

- `search_paraula()`

L'objectiu d'aquest test és comprovar si la funció de cercar paraules al trie funciona correctament. Entenem que una paraula al trie és un string que l'última lletra apunta a un node que el seu booleà `endOfWord` és true.

Creem un trie al qual afegim el nom d'un autor i busquem amb la funció `search` aquesta paraula al trie. Observem que el booleà que ens retorna és true i per tant troba la paraula al trie.

- `search_prefix()`

L'objectiu d'aquest test és comprovar si la funció de cercar prefixos al trie funciona correctament. Entenem que un prefix al trie és un string que es troba a l'estructura, però l'últim node no apunta necessàriament a un node on el seu `endOfWord` sigui true.

Creem un trie al qual afegim el nom d'un autor, per exemple "neus", i busquem amb la funció *search_prefix* la primera lletra del nom, és a dir, "n". Observem que el booleà que ens retorna la funció és true i per tant troba el prefix al trie sense que necessàriament sigui una paraula.

- *cercaeliminada()*

L'objectiu d'aquest test és comprovar si un cop eliminat un autor del trie, si el tornem a buscar a l'estructura ens retorna false, i per tant ha sigut eliminat correctament.

En primer lloc, creem un trie al qual afegim el nom d'un autor i el busquem a l'estructura amb la funció *search*. Observem que ens retorna un booleà true i per tant ha trobat correctament la paraula.

En segon lloc, eliminem aquest autor del trie i el tornem a buscar i veiem que efectivament aquest cop el booleà ens retorna false.