

Tercera pràctica d'Estructures de dades

Índex alfabètic d'un text

Disposem d'un fitxer de text i volem crear un índex amb les paraules que són més representatives i on han aparegut cada una d'elles dins el text. Una paraula la considerarem representativa si, en la primera vegada que apareix en el fitxer, té afegit en la primera posició el caràcter \$. En la resta d'aparicions, la paraula, no portarà aquest caràcter marcador, però també s'ha d'incloure en l'índex. La sortida al programa serà l'índex amb les paraules representatives ordenades alfabèticament. I de cada paraula s'indicarà en quines planes i línies del fitxer de text ha aparegut.

Per a identificar la plana en que ens trobem en el fitxer de text considerarem una marca especial <Plana numero=i> on i serà el valor de plana en què ens trobem a partir d'aquest moment i fins que no trobem la següent marca on s'indica la plana següent. Per a identificar en quina línia ens trobem dins de la plana, haurem d'anar comptant quantes línies de text portem llegides en ella.

Un fitxer exemple d'entrada podria ser:

<Plana numero=1>

Un \$arbre \$binari és una estructura de dades en la que cada \$node sempre té un fill esquerre i un fill dret. L'arbre binari te diferents \$especificacions particulars, l'arbre binari de \$cerca, el \$monticle, entre altres.

<Plana numero=2>

El monticle binari és un cas particular on tot node té un valor superior al dels seus fills i a més compleix la propietat de que és un arbre \$complet.

<Plana numero=3>

Un arbre és \$equilibrat si per a qualsevol node la diferència d'alçades entre els seus dos fills és inferior a 2.

Un arbre és complet quan està equilibrat i totes les fulles del nivell inferior es troben a la part esquerra de l'arbre.

I l'índex alfabètic que s'hauria de generar seria:

arbre 1:1, 1:2, 2:2, 3:1, 3:3, 3:4

binari 1:1, 1:2, 1:3, 2:1

cerca 1:3

complet 2:2, 3:3

especificacions 1:2

equilibrat 3:1, 3:3

monticle 1:3, 2:1

node 1:1, 2:1, 3:1

Una forma d'indicar que una paraula ha sortit a la plana X, línia Y, en l'índex, pot ser X:Y. Si una mateixa paraula surt dos cops en una línia només s'ha d'indicar una vegada.

A l'entorn Moodle trobareu aquest fitxer exemple però per als jocs de proves n'heu de dissenyar altres que contemplin casos reals.

Volem resoldre el problema amb dos versions:

- Una primera utilitzant l'estructura de dades Taula de dispersió i
- Una segona amb una estructura de dades basada en Arbres. El tipus d'arbre es deixa a l'elecció del grup a la que es consideri més adient.

De cada versió de TAD utilitzat **es demana** només una implementació i es valorarà la idoneïtat de l'escollida segons el tipus de problema plantejat. Opcionalment es pot complementar la implementació utilitzant estructures de les *java.util*

Es tracta de dissenyar els TADs més adients per a guardar la informació de la millor forma i que donin resposta a les necessitats descrites.

S'han de fer els TADs el més genèrics possibles, definir i utilitzar iteradors per recórrer els elements que formen l'estructura en el main i implementar/testejar les operacions de consulta, afegit i esborrat.

Informe de la pràctica

Es demana un informe que contingui:

- una breu explicació dels TADs escollits i les implementacions triades. Justificació de cada decisió.
- anàlisi i comparació del cost temporal i espacial de les operacions en les dues versions del TAD, taules de dispersió i arbres.
- anàlisi dels temps d'execució dels diferents jocs de proves comparant amb els diferents TADs.

Instruccions per al desenvolupament de la pràctica

- Aquesta pràctica es farà en grups de dos persones.
- Heu de crear el *Java Project* d'Eclipse amb el nom, **ED_PRACX_COGNOM1_NOM1_COGNOM2_NOM2**, corresponents als noms dels dos integrants del grup.
- Lliurareu els codis fonts (fitxer Java) i un informe en pdf segons les indicacions que hi ha en aquest enunciat. Tota aquesta informació es guardarà com a projecte exportat de l'Eclipse (fitxer ZIP) i es penjarà a la tasca corresponent del "moodle". El nom del fitxer serà **ED_PRACX_COGNOM1_NOM1_COGNOM2_NOM2.zip**
- El termini de lliurament és el divendres **02 de juny de 2017 a les 21h**.
- En cap cas es consideraran lliurades les pràctiques que no arribin amb el procés indicat (codi + informe).
- No s'acceptaran pràctiques que (a) No estiguin acabades. (b) No facin el que es demana a l'enunciat. (c) No compilin i executin. o (d) No compleixin amb la nomenclatura d'entrega.
- No es poden fer servir col·leccions de Java (`java.util.*`) excepte en la part opcional.