

PAC 2

Exercici 1 [2 punts]

Apartat 1.1) Descriviu com funcionaria un algorisme que us permeti fusionar dos arbres AVL de mida N i M respectivament. Quin cost tindria aquesta operació de fusió?

Apartat 1.2) Implementeu amb Java una extensió de la classe `ArbreAVL` de la biblioteca de classes per dotar-la d'un mètode que fusioni l'arbre actual amb un arbre que rebeu com a paràmetre. El mètode que cal que implementeu ha de tenir la següent signatura:

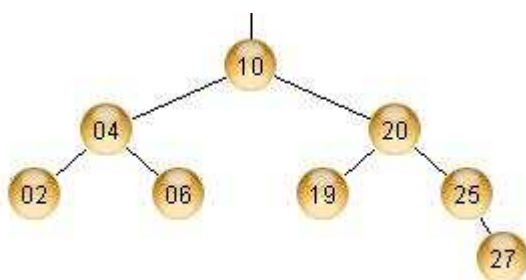
```
public void fusionar(ArbreAVL arbre)
```

Apartat 1.3) Apliqueu l'algorisme de fusió per fusionar els elements de l'arbre X amb l'arbre Y i mostreu l'arbre AVL resultant. Si la operació de fusió provoca desequilibris, comenteu quin desequilibri es produeix i amb quina rotació es soluciona.

Arbre X



Arbre Y



Exercici 2 [2 punts]

Apartat 2.1) Implementeu amb Java un TAD que us permeti gestionar els empleats d'una empresa multinacional amb les següents premisses:

- No podeu utilitzar delegació.
- Utilitzeu les classes de la JCF.
- El nombre d'empleats és molt gran i conegut.
- Cal un accés ràpid a les dades de l'empleat a partir del seu dni.
- Per cada empleat guardarem les següents dades:
 - Dni
 - Nom
 - Cognoms
 - Telèfon
- Cal que ofereixi operacions per:
 - Donar d'alta un empleat
 - Consultar l'empleat per dni
 - Donar de baixa un empleat
 - Consultar si un dni correspon a un empleat
 - Llistar tots els empleats
 - Llistar tots els dni dels empleats
 - Consultar el nombre d'empleats
 - Consultar el nombre d'empleats amb un nom determinat
 - Afegir un conjunt d'empleats

Apartat 2.2) Implementeu amb Java un TAD que us permeti gestionar els noms dels empleats d'una empresa multinacional amb les següents premisses:

- No podeu utilitzar herència.
- Utilitzeu les classes de la biblioteca de classes de l'assignatura.
- El nombre d'empleats és molt gran però desconegut.
- Cal una consulta eficient.
- Cal que ofereixi operacions per:
 - Donar d'alta un nom, si el nom ja existeix no fa res
 - Consultar si un nom existeix
 - Eliminar un nom
 - Llistar tots els noms
 - Consultar el nombre de noms que comencin per 'A'
 - Afegir un conjunt de noms



Exercici 3 [2,5 punts]

Apartat 3.1) Suposem que hem dissenyat una funció de dispersió que té com a claus els noms i cognoms dels estudiants matriculats a la UOC; suposem també que entre nom i cognoms es poden ocupar 60 caràcters. Es defineix la funció de dispersió de la següent manera:

$$h(c_1 \dots c_{60}) = (\sum_{k: 1 \leq k \leq 60} \text{ascii}(c_k) * 2^k) \bmod 256$$

Comenta els possibles inconvenients de la funció de dispersió escollida i proposa una alternativa que els solucioni.

Apartat 3.2) És cert que un TAD taula amb una funció de dispersió mal dissenyada pot degradar el rendiment de l'operació de cerca fins a fer-la lineal respecte del nombre d'elements de la taula? Posa un exemple de funció de dispersió amb aquest comportament.

Apartat 3.3) Suposeu que teniu un TAD taula i que no aconseguiu una funció de dispersió que doni una distribució equiprobable dels elements, com podríeu millorar l'eficiència de la cerca sense variar la funció de dispersió?



Exercici 4 [3,5 punts]

Hem d'implementar un prototip per realitzar votacions per Internet experimentant un sistema electoral amb llistes obertes. En aquest sistema els candidats pertanyen a algun partit però els electors no voten a un partit, sinó a un candidat concret.

Un cop realitzades les votacions es fa la divisió entera entre el nombre total de votants i els E escons, obtenint així la *quota* necessària per obtenir un escó. Tots els candidats que estan per sobre de la *quota* obtenen escó, i els que estan per sota queden provisionalment fora.

Si el procés s'acaba aquí es molt probable que no s'ocupin tots els E escons, ja que segurament alguns dels candidats poden haver superat la *quota* i un candidat només pot ocupar un escó, encara que l'hagin votat suficientment per ocupar-ne dos o més. Per tal d'ocupar tots els E escons, cal redistribuir aquest excedent de *quota*, però ¿a qui? Nosaltres ens interessem per la següent solució:

En el moment de votar, es demana als electors que ordenin els candidats de més a menys prioritari. D'aquesta manera, l'excedent de vots d'un candidat X es pot redistribuir d'acord amb la voluntat dels electors. Normalment, però, els electors només ordenen uns pocs candidats (empíricament se sap que amb 10 candidats ja és suficient per garantir el procés).

El procediment de redistribució d'excedent és el següent: Per cada candidat C amb excedent, es divideix l'excedent entre els seus votants NV, obtenint l'excedent proporcional (un valor decimal). És a dir, cada votant del candidat C "gasta" una part pel candidat C i, la resta, la dedica al següent candidat que hagi triat. Cada cop que es redistribueix l'excedent d'un candidat, aquest passa al grup de candidats seleccionats i ja no intervindrà més en el procés de redistribució. Aquest procediment es va repetint fins que desapareix l'excedent de tots els candidats.

Per exemple, si suposem que tenim 100 vots, 4 escons, 5 candidats (C1, C2, C3, C4 i C5) i que a les votacions C1 obté 50 vots, C2 n'obté 35 i C3 n'obté 15 els altres dos no obtenen cap vot. Amb aquestes dades tenim:

- Quota: 25 (100 vots / 4 escons)
- C1 té un excedent de quota de 25 (50-25) i un excedent proporcional de $(25/50) = 0,5$. El 50% del vot dels electors del candidat C1 s'acumula a la seva segona opció.
- C2 té un excedent de quota de 10 (35-25) i un excedent proporcional de $(10/35) = 0,28$. El 28% del vot dels electors del candidat C2 s'acumula a la seva segona opció.
- C3, C4 i C5 no tenen excedent de quota.

El procediment de redistribució d'excedents es faria de la manera següent:

- Triar un dels candidats amb excedent (suposem Cx).
- Recórrer tots els vots del candidat i, per cada un dels vots
 - Triar el primer candidat del vot (recordeu que en el moment de votar es demana als electors que ordenin de major a menor els candidats als que donen el seu vot) que no hagi entrar encara en el procediment de redistribució (suposem que és Cy)
 - Afegir el vot a Cy (actualitzant la fracció del vot amb l'excedent proporcional de Cx)



- Actualitzar el nombre de vots de Cy (suma de les fraccions de tots els vots de Cy)
- Comprovar si Cy supera la quota
- Actualitzar el nombre de vots de Cx (ara seran igual a la quota perquè hem redistribuït el seu excedent)
- Marcar que el candidat Cx ja ha entrat al procediment de redistribució.

Un cop redistribuïts tots els excedents, començaria una segona etapa d'eliminació de candidats fins quedar-ne E (el nombre d'escons), però de moment no implementarem aquesta segona etapa.

Per la resolució de l'exercici fem les següents consideracions:

- El nombre de partits P és petit i conegut.
- El nombre de candidats C és relativament gran i conegut (de l'ordre de milers).
- El nombre de votants V és desconegut (depèn del grau de participació) i pot ser molt gran (de l'ordre de milions).
- El nombre d'escons E és petit i conegut.

Es vol dissenyar un TAD Votacions amb les operacions següents:

- 1) crear(). Crear l'estructura, inicialment buida.
- 2) afegirPartit(Partit). Afegeix un partit a l'estructura.
- 3) afegirCandidat(Candidat, Partit). Afegeix un candidat associat a un partit.
- 4) afegirVotant(Votant, cua <Candidat>). Registra un votant amb la seva llista de preferències i actualitza el total de vots del candidat en primera opció. Si el votant ja havia votat, dona un missatge d'error. Se suposa que la llista de candidats és correcta.
- 5) finalitzarVotacions(). Tanca les votacions i prepara l'estructura pel procés de redistribució.
- 6) quedenExcedents(). Retorna un booleà indicant si queden candidats amb excedent > 0 .
- 7) redistribuirExcedent(). Redistribueix l'excedent del candidat amb més excedent.
- 8) llistatCandidatsSeleccionats(). Retorna un iterador per recórrer els candidats seleccionats. No importa l'ordre. Inicialment aquesta llista és buida i es va ampliant a cada pas del procés de redistribució.
- 9) partitMésVotat(). Retorna el partit més votat.

Requisits d'eficiència:

- Les operacions 1, 2, 3 i 5 no cal que siguin especialment eficients.
- Les operacions 4, 6 i 7 han de ser el màxim d'eficients possibles.



PAC2 Estructura de la Informació curs 2010/2011 1r semestre de FUOC està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

Us demanem el següent:

Apartat 4.1) Feu un dibuix de l'estructura de dades resultant, que deixi clares les parts que la componen mitjançant les representacions gràfiques vistes a l'assignatura. Podeu posar una breu descripció (dues o tres línies) de cada component de l'estructura (per exemple, l'estratègia de representació: seqüencial, encadenada per vectors, encadenada indirecte, etc.). Ha de quedar clar quina és la informació continguda a cadascun d'aquests components.

Apartat 4.2) Estudieu l'eficiència de les operacions `afegirVotant()` i `redistribuirExcedent()`. Concretament, per a cada operació heu de descriure breument el seu comportament indicant els passos que la componen (amb frases com ara: “inserir en l'arbre AVL / esborrar de la taula de dispersió / consulta del piló / ordenar el vector...”), dient l'eficiència asimptòtica de cada pas i donant l'eficiència total de l'operació.



PAC2 Estructura de la Informació curs 2010/2011 1r semestre de FUOC està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)