

PAC2 – Segona Prova d'Avaluació Continuada

Presentació

A continuació us presentem l'enunciat de la segona prova d'avaluació continuada del curs. Tingueu en compte que la PAC s'ha de resoldre individualment.

Competències

Les competències que treballareu a la PAC són les següents:

Específiques

- Capacitat per analitzar un problema en el nivell d'abstracció adequat a cada situació i aplicar les habilitats i coneixements adquirits per a resoldre'l.
- Capacitat per dissenyar i construir aplicacions informàtiques mitjançant tècniques de desenvolupament, integració i reutilització.
- Capacitat per proposar i avaluar diferents alternatives tecnològiques per resoldre un problema concret.

Transversals

- Ús i aplicació de les TIC en l'àmbit acadèmic i professional.
- Capacitat per a innovar i generar noves idees.

Objectius

Els objectius que es persegueixen en el desenvolupament de la PAC són els següents:

- Entendre el concepte de TAD i saber-ne fer l'especificació.
- Conèixer la biblioteca de TADs de l'assignatura i saber utilitzar-los per dissenyar i implementar noves estructures de dades.
- Saber calcular l'eficiència espacial i temporal d'una estructura de dades i dels algorismes associats per tal de comparar diferents alternatives i poder triar-ne la millor en termes d'eficiència (temporal o espacial)
- Ser capaç d'identificar l'estructura de dades utilitzada en un programa i entendre'n el funcionament.
- Entendre el funcionament dels contenidors seqüencials i els arbres presentats a l'assignatura. Saber quan i com utilitzar aquests contenidors.

Descripció de la PAC a realitzar

La PAC consta de 4 exercicis, alguns més teòrics i altres més pràctics, en els que l'alumne posarà en pràctica els coneixements adquirits en l'estudi de quatre mòduls de l'assignatura.



Recursos

Els recursos necessaris per a desenvolupar la PAC son els següents:

Bàsics (material didàctic de l'assignatura)

- Mòdul 4
- Mòdul 5
- Mòdul 6
- Mòdul 7

Complementaris

No es requereixen materials complementaris.

Criteris de valoració

La puntuació global de la PAC és la suma de les puntuacions individuals obtingudes a cada un dels exercicis que la formen. La puntuació individual de cada exercici s'especifica a cada un d'ells.

Format i data de lliurament

Data de publicació

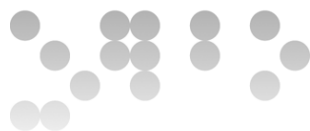
18 de novembre de 2013

Data de lliurament

2 de desembre de 2013

Format de lliurament

El lliurament cal fer-lo a través de l'espai de Lliurament i registre d'AC amb un únic fitxer preferentment en format PDF o, si no és possible, Word o OpenOffice. Aquest fitxer contindrà la solució (incloent-ne les classes Java si s'escau). Si us plau, no hi copieu l'enunciat, feu-hi constar el vostre nom a cada pàgina (per exemple, amb un peu de pàgina), i numereu les pàgines.



Enunciat

En aquesta PAC continuarem dissenyant una estructura de dades per gestionar el servei de *Bicing*. Ara ampliarem les funcionalitats i treballarem amb volums d'informació grans que requeriran la utilització d'estructures de dades dels mòduls 4 al 7. Concretament, per a la realització de l'exercici considereu:

De la PAC1:

- El nombre de places d'una estació P continua petit i conegut en el moment de donar d'alta l'estació.
- El nombre de bicicletes aparcades en una estació BS és petit però molt variable.
- El nombre de places de pàrquing de totes les estacions és força més gran que B, per garantir poder aparcar les bicicletes amb certa facilitat.
- El nombre de serveis realitzats per una bicicleta BU serà gran i anirà en constant augment.
- El nombre de serveis realitzats per un usuari UU serà petit però anirà en augment.

Canvis respecte la PAC1:

- El nombre d'estacions de bicicletes S augmenta força, però continua essent conegut.
- El nombre de bicicletes B continua conegut, però augmenta molt, de l'ordre d'uns quants milers.
- El nombre d'usuaris U continua variable però ara el considerarem molt gran, de l'ordre de centenars de milers.

Nou a la PAC2:

- El nombre d'operaris W és gran i conegut.
- El nombre d'incidències I es preveu molt gran i en constant augment.
- El nombre d'incidències d'una bicicleta BI serà petit, però molt variable.
- El nombre d'incidències d'un operari WI serà més gran i en constant augment.



Exercici 1 [2'5 punts]

Especifiqueu un TAD *Bicing* per permeti:

De la PAC1:

- Afegir una nova estació al sistema a partir del seu codi identificador (natural), coordenades geogràfiques (latitud i longitud), i el nombre de pàrquings de bicicletes que tindrà.
- Afegir una nova bicicleta. Caldrà especificar el codi identificador de la bicicleta (string amb la matrícula), el codi de l'estació i el número de plaça de pàrquing on s'aparcara inicialment.
- Afegir un nou usuari al sistema. De cada usuari en sabrem el seu codi identificador (natural) i el seu nom.
- Obtener una bicicleta lliure d'una estació per a un usuari. El sistema buscarà la bicicleta menys utilitzada, de les que estan aparcades a l'estació en aquell moment.
- Registrar que una bicicleta ha estat aparcada (i, per tant, retornada) a una estació determinada.
- Consultar els serveis realitzats per una bicicleta determinada en ordre cronològic.
- Consultar quina és l'estació més utilitzada.
- Consultar quin és l'usuari que ha agafat més cops la bicicleta.

Nou a la PAC2:

- Afegir un nou operari. De cada operari en sabrem el seu codi identificador (string amb el DNI), i el seu nom.
- Registrar una nova incidència d'una bicicleta. Sabrem l'identificador de la bicicleta, una descripció de l'avaria, i la data/hora. El sistema retornarà l'identificador de la incidència (un natural). La incidència estarà en estat *pendent d'assignar*.
- Assignar una incidència a un operari. Rebrem l'identificador de la incidència, i la data/hora de l'assignació. S'assignarà a l'operari amb menys incidències assignades, i la incidència passarà a estat *assignada*. Retorna l'operari assignat.
- Resoldre una incidència. Rebrem l'identificador de la incidència, una explicació de la reparació i la data/hora. La incidència passarà a estat *resolt*.
- Consultar les incidències d'una bicicleta.
- Consultar les incidències d'un operari. Aquesta operació pot ser lineal respecte el nombre d'incidències l.
- Consultar quina és l'estació més conflictiva (la que ha registrat més incidències). Fixeu-vos que en registrar una nova incidència només sabem l'identificador de la bicicleta, no de l'estació. Però s'entén que, en el moment de registrar-se la incidència, la bicicleta estava aparcada en una estació.
- Consultar el nombre de bicicletes aparcades en una estació (que no tinguin una incidència pendent de resoldre).



- Consultar el nombre de places lliures en una estació.
- Consultar l'estació més propera amb una o més bicicletes lliures a partir d'unes coordenades (latitud, longitud). Aquesta operació pot ser lineal respecte el nombre d'estacions.
- Consultar l'estació més propera amb una o més places lliures a partir d'unes coordenades (latitud, longitud). Aquesta operació també pot ser lineal respecte el nombre d'estacions.

Apartat a) [1 punt]

Dóna la signatura del TAD *Bicing* dels mètodes nous de la PAC2. És a dir, indica el nom que donaries a les operacions encarregades de cada funcionalitat requerida. Indica, també, quins caldria que fossin els paràmetres d'entrada i quina la sortida en cas que es necessités.

Solució:

```
void addWorker(DNI, name);
int void addTicket(bicycleID, description, datetime);
Worker assignTicket(ticketID, datetime);
void resolveTicket(ticketID, description, datetime);
Iterator ticketsByBicycle(bicycleID);
Iterator ticketsByWorker(workerID);
Station mostProblematicStation();
Int getNBicycles(stationID);
Int getNParkings(stationID);
Station getClosestBike(latitude, longitude);
Station getClosestParking(latitude, longitude);
```

Apartat b) [1,5 punts]

Fes l'especificació contractual de les operacions del TAD *Bicing*, modificades o noves a la PAC2. En la redacció de l'especificació pots fer servir, si et cal, qualsevol de les operacions del TAD. Pren com a model l'especificació de l'apartat 1.2.3 del Mòdul 1 dels materials docents. Es valorarà especialment la concisió (absència d'elements redundants o innecessaris), precisió (definició correcta del resultat de les operacions), completeness (consideració de tots els casos possibles en què es pot executar cada operació) i manca d'ambigüitats (coneixement exacte de com es comporta cada operació en tots els casos possibles) de la solució. És important respondre aquest apartat usant una descripció condicional i no procedimental. L'experiència ens demostra que no sempre resulta fàcil distingir entre les dues descripcions, és per això que fem especial èmfasi insistint que poseu molta atenció a les vostres definicions.



A títol d'exemple indicarem que la descripció condicional (la correcta a utilitzar en el contracte) d'omplir un got buit amb aigua seria:

```
@pre el got es troba buit.
@post el got és ple d'aigua.
```

En canvi una descripció procedimental (incorrecta per utilitzar en el contracte) tindria una forma semblant a:

```
@pre el got hauria de trobar-se buit, si no es trobés buit s'hauria de buidar.
@post s'acosta el got a l'aixeta i s'hi tira aigua fins que estigui ple.
```

Cal també tenir en compte que un contracte hauria de disposar d'invariant sempre que aquesta fos necessària per descriure el TAD.

Solució:

```
@pre no existeix un operari amb el DNI.
@post els operaris seran els mateixos més un de nou amb les dades indicades.
void addWorker(DNI, name);
```

```
@pre cert
@post els tiquets seran els mateixos més un nou tiquet amb les dades indicades i estat pendent d'assignar. Retorna l'identificador d'aquest nou tiquet.
int void addTicket(bicycleID, description, datetime);
```

```
@pre existeix un tiquet amb identificador ticketID amb estat pendent d'assignar i, com a mínim, un operari donat d'alta.
@post els tiquets seran els mateixos exceptuant el tiquet identificat per ticketID, que haurà canviat a estat assignat, tindrà operari assignat (el que tingui assignades menys incidències) i data/hora de l'assignació. Retorna l'operari assignat.
Worker assignTicket(ticketID, datetime);
```

```
@pre existeix un tiquet amb identificador ticketID amb estat assignat
@post els tiquets seran els mateixos exceptuant el tiquet identificat per ticketID, que haurà canviat a estat resolt, i en sabrem la reparació feta i data/hora de la reparació
void resolveTicket(ticketID, description, datetime);
```

```
@pre existeix una bicicleta identificada per bicycleID
@post retorna un iterador per poder recórrer tots els tiquets de la bicicleta.
Iterator ticketsByBicycle(bicycleID);
```



@pre existeix un operari identificat per workerID

@post retorna un iterador per poder recórrer tots els tiquets resolts per l'operari.

Iterator ticketsByWorker(workerID);

@pre cert

@post retorna l'estació on s'han registrar més tiquets.

Station mostProblematicStation();

@pre existeix una estació identificada per stationID

@post retorna el nombre de bicicletes lliures que no tinguin tiquets pendents de resoldre.

int getNBicycles(stationID);

@pre existeix una estació identificada per stationID

@post retorna el nombre de places lliures de l'estació.

int getNParkings(stationID);

@pre cert

@post retorna l'estació més propera a les coordenades latitude, longitude que tingui una o més bicicletes lliures, sense tiquets pendents de resoldre.

Station getClosestBike(latitude, longitude);

@pre cert

@post retorna l'estació més propera a les coordenades latitude, longitude amb places lliures.

Station getClosestParking(latitude, longitude);

Exercici 2 [3,5 punts]

A l'Exercici 1 heu definit l'especificació del TAD *Bicing*. Ara us demanem que feu el disseny de les estructures de dades que formaran aquest TAD. Dissenyeu, doncs, el sistema per tal que sigui el màxim d'eficient possible, tant a nivell d'eficiència espacial com temporal, tenint en compte els volums d'informació i les restriccions especificades a l'enunciat.

Tingueu en compte només les operacions que es demanen a l'enunciat a l'hora de fer aquest disseny.

Apartat a) [0,5 punts]

Dubtem entre utilitzar una taula de dispersió, un AVL o continuar amb un vector (com a la PAC1) per emmagatzemar les estacions. Justifiqueu quina creieu que és la millor opció.



Solució: continuar amb Vector

El nombre d'estacions, tot i augmentar, continua essent conegut. Descartem els AVL ja que ens portarien costos logarítmics mentre que les taules de dispersió o els vectors ens permeten costos constants. Si els identificadors de les estacions són continus, els vectors continuen essent la millor solució (costos constants però sense càlcul de funció de dispersió, ni mètode de redispersió,...i molt més simple d'implementar).

Apartat b) [0,75 punts]

Dubtem entre utilitzar una taula de dispersió, un AVL o una cua prioritària per emmagatzemar els operaris. Justifiqueu quina creieu que és la millor opció.

Solució: cua prioritària

Als operaris no hi hem d'accedir-hi mai per identificador. Només necessitem accedir a l'operari amb menys incidències assignades. Ni les taules de dispersió, ni els AVL estan preparats per aquesta funcionalitat. Així doncs, triem una cua prioritària, que ens permetrà obtenir l'operari en cost constant, i fer les actualitzacions en cost logarítmic.

Apartat c) [0,5 punts]

Dubtem entre utilitzar una cua prioritària, una llista encadenada o un AVL per emmagatzemar les incidències. Justifiqueu quina creieu que és la millor opció.

Solució: AVL

Les cues prioritàries permeten ordenar els elements per una prioritat. Però les incidències no tenen cap prioritat assignada per tant, en aquest cas, no tindria sentit. El nombre d'incidències va en constant augment i tant les llistes encadenades com els AVL anirien bé. Però a diferents mètodes del TAD Bicing cal accedir a les incidències a partir del seu identificador. Les llistes comportarien costos lineals i, per això, triem un AVL.

Apartat d) [0,5 punts]

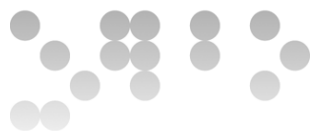
Dubtem entre utilitzar un llista encadenada, un AVL o una taula de dispersió emmagatzemar les incidències assignades a un operari. Justifiqueu quina creieu que és la millor opció.

Solució: res

L'enunciat ens diu que aquesta operació pot ser lineal respecte el nombre d'incidències. Això ens permet fer un recorregut lineal sobre l'AVL d'incidències i filtrar les que siguin de l'operari en qüestió.

Apartat e) [0,5 punt]

Dubtem entre utilitzar un AVL, una taula de dispersió o continuar amb un vector ordenat per emmagatzemar les bicicletes. Justifiqueu quina creieu que és la millor opció.

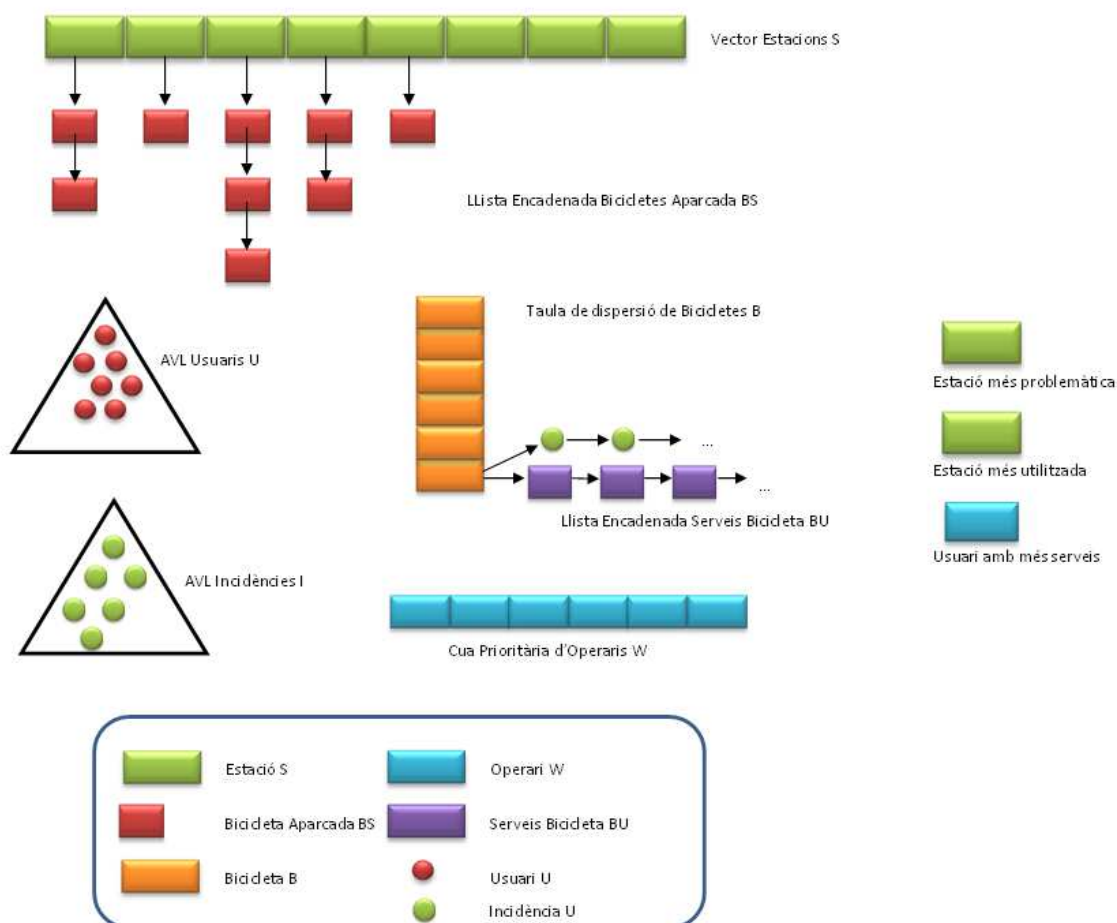


Solució: Taula de dispersió

El nombre de bicicletes ha augmentat molt però continua constant. Això ens permet utilitzar qualsevol de les estructures de dades proposades. Continuar amb el vector ordenat o triar un AVL permet fer consultes logarítmiques, però el vector ordenat penalitza les insercions. Malgrat tot, és la taula de dispersió l'estructura més eficient (amb costos constants) i, com que no necessitem un llistat ordenat de bicicletes, serà l'opció triada.

Apartat f) [0,75 punt]

Feu un dibuix de l'estructura de dades global pel TAD *Bicing* on es vegin clarament les estructures de dades que trieu per representar cada una de les parts i les relacions entre elles. Cal que feu el dibuix de l'estructura complerta, amb totes els estructures que us permetin implementar les operacions definides a l'especificació.





Exercici 3 [3 punts]

A l'Exercici 1 heu definit l'especificació del TAD *Bicing* amb les seves operacions i a l'Exercici 2 heu triat les estructures de dades per cada part del TAD. En aquest exercici us demanem que us fixeu en els algorismes que us serviran per implementar algunes de les operacions especificades i en l'estudi d'eficiència de les mateixes. Tingueu en compte que la implementació de les operacions va estretament lligat a l'elecció de les estructures de dades que hagueu fet.

Apartat a) [2 punts]

Descriviu i feu l'estudi d'eficiència de les operacions que hagueu definit per crear noves incidències, assignar-les a operaris, i registrar la resolució considerant:

Considerem que:

- una Incidència tindrà un atribut que serà una referència a la Bicicleta
- una Bicicleta tindrà un atribut que serà una referència a l'Estació on està aparcada.
- una Incidència tindrà un atribut que serà una referència a l'Operari assignat.

Per fer-ho 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. No heu de fer el pseudocodi, només descriure els passos.

Solució:

Nova Incidència:

- Cercar la bicicleta a la taula de dispersió de bicicletes $O(1)$
- Modificar l'estat de la bicicleta a "avariada" $O(1)$
- Crear un objecte Incidència a partir dels paràmetres i la bicicleta $O(1)$
- Afegir l'objecte a l'AVL d'incidències $O(\log I)$
- Consultar l'estació on està aparcada la bicicleta $O(1)$
- Incrementar el nombre d'incidències de l'estació $O(1)$
- Actualitzar l'estació més conflictiva, si s'escau $O(1)$

Total: $O(\log I)$

Assignar Incidència a Operari:

- Cercar la incidència a l'AVL d'incidències $O(\log I)$
- Modificar l'estat i hora de la incidència $O(1)$
- Consultar el primer operari de la cua prioritària d'operaris, és a dir, el que menys incidències té assignades $O(1)$
- Incrementar el nombre d'incidències assignades de l'operari $O(1)$
- Actualitzar l'operari a la cua de prioritat (enfonsar-lo) $O(\log W)$
- Assignar l'operari a la incidència $O(1)$

Total: $O(\log I + \log W)$



Resoldre Incidència:

- Cercar la incidència a l'AVL d'incidències $O(\log I)$
- Modificar l'estat, explicació i hora de la incidència $O(1)$
- Consultar la bicicleta de la incidència $O(1)$
- Modificar l'estat de la bicicleta a "en servei" $O(1)$

Total: $O(\log I)$

Apartat b) [1 punts]

Repetiu l'exercici anterior considerant que l'operació d'assignar operari a una incidència, en comptes d'assignar l'operari amb menys incidències assignades, o faci a l'operari amb menys incidències resoltes.

Solució:

Nova Incidència: queda igual

Assignar Incidència a Operari:

- Cercar la incidència a l'AVL d'incidències $O(\log I)$
- Modificar l'estat i hora de la incidència $O(1)$
- **Consultar el primer operari de la cua prioritària d'operaris, és a dir, el que menys incidències té resoltes $O(1)$**
- **Assignar l'operari a la incidència $O(1)$**

Total: $O(\log I)$

Resoldre Incidència:

- Cercar la incidència a l'AVL d'incidències $O(\log I)$
- Modificar l'estat, explicació i hora de la incidència $O(1)$
- Consultar la bicicleta de la incidència $O(1)$
- Modificar l'estat de la bicicleta a "en servei" $O(1)$
- **Consultar l'operari de la incidència $O(1)$**
- **Incrementar el nombre d'incidències resoltes de l'operari $O(1)$**
- **Cercar l'operari a la cua de prioritat $O(W)$**
- **Actualitzar la posició de l'operari a la cua de prioritat (enfonsar-lo) $O(\log W)$**

Total: $O(\log I + W + \log W)$



Exercici 4 [1 punt]

Indica quins dels TADs de la biblioteca de TADs de l'assignatura et semblen més adients per utilitzar-los en la implementació de cada una de les estructures de dades definides pel TAD *Bicing*.

Solució:

- Per les estacions continuem amb un array de java.
- Pels tiquets i usuaris triem una *DiccionariAVLImpl*
- Per les bicicletes triem una *TaulaDispersio*.
- Pels operadors triem una *CuaAmbPrioritat*.
- Per les bicicletes aparcades a una estació, i pels serveis d'una bicicleta continuem amb una *LlistaEncadenada*.
- Per les incidències d'una bicicleta utilitzem una *LlistaEncadenada*.