Ús de Bases de Dades

PAC 4: Implementació de mètodes d'accés.

Presentació

En aquesta Prova d'Avaluació Continuada s'exerciten els aspectes que convé tenir en compte en el disseny i implementació dels índexs d'una base de dades. L'objectiu d'aquesta prova és comprovar el grau de comprensió del mòdul 4. Aquesta prova consta de 4 exercicis. Cal destacar que és necessari haver assimilat el contingut del mòdul 4 per a la correcta realització d'aquesta PAC.

Competències

En aquesta PAC es desenvolupen les següents competències del Grau en Multimèdia:

- Capacitat d'analitzar un problema en el nivell d'abstracció adequat a cada situació i aplicar les habilitats i coneixements adquirits per abordar-lo i resoldre'l.
- Capacitat per aplicar les tècniques específiques de tractament, emmagatzematge i administració de dades.

Objectius

Els objectius concrets d'aquesta Prova d'Avaluació Continuada són:

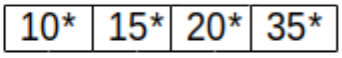
- Conèixer els diferents mètodes d'accés que són necessaris per a poder fer consultes i actualitzacions a les dades emmagatzemades a les BD.
- Entendre la importància que té la reducció del nombre d'E/S en les implementacions dels mètodes d'accés.
- Comprendre la utilitat dels índexs per a la implementació dels accessos per valor.
- Conèixer l'estructura dels índexs arbres B+ .
- Conèixer els índexs organitzats amb funcions de dispersió.
- Saber quines són les característiques dels índexs agrupats.
- Entendre els avantatges i inconvenients dels índexs arbres B+, dels organitzats amb funcions de dispersió i dels índexs agrupats amb vista a la implementació dels accessos per valor.
- Conèixer els índexs de valors compostos i entendre'n els avantatges i els inconvenients per a la implementació dels accessos per diversos valors.

Exercici 1 [25%]

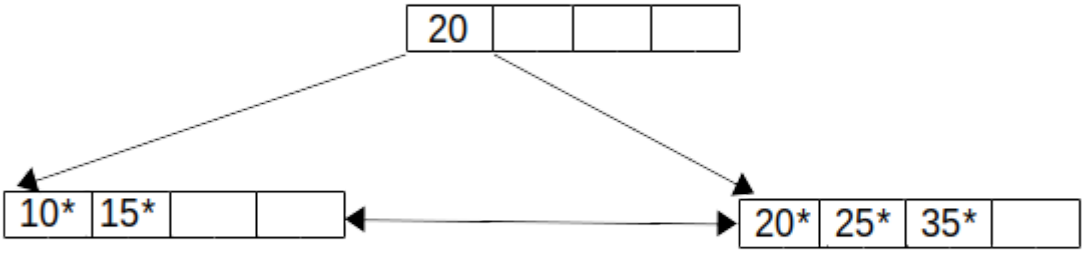
Partint d'un arbre B+ d'ordre 2 buit insereix consecutivament els valors **15**, **20**, **35**, **10**, **25**, **44**, **7**, **65**, **8**, **75** i **14** Realitza les operacions indicades, explica els criteris que has emprat en cada operació i mostra tots els arbres intermedis resultants de cada inserció i el resultat final.

Proposta de solució:

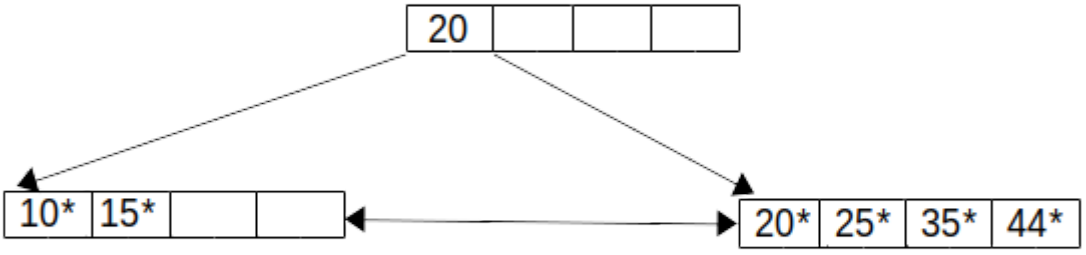
Inserim els valors 15, 20, 35 i 10.
Partim d'un arbre B + d'ordre d = 2, per tant la capacitat màxima de cada node serà de 4 valors (2d=4) per als nodes interns o 4 entrades per als nodes fulla. En conseqüència, el primer node podrà albergar les entrades corresponents als quatre primers valors, tenint la precaució d'ordenar després de cada inserció:



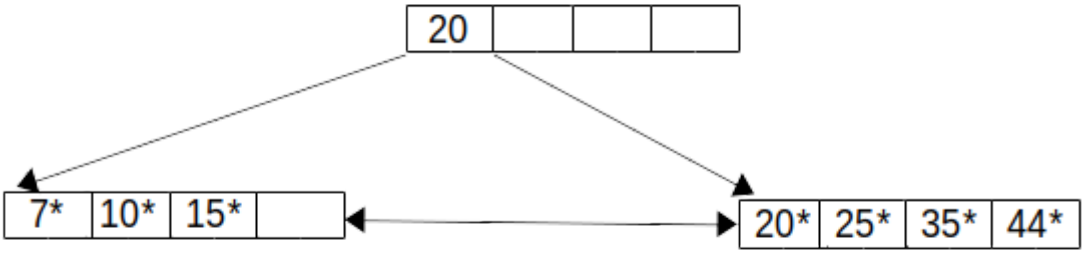
Inserim 25.
El node anterior no disposa d'espai lliure per a albergar una nova entrada, de manera que per inserir-la cal dividir el contingut del node en dos. Dels dos nodes producte de la divisió, el primer albergarà les *d* primeres entrades (10 i 15) i el segon les *d+1* últimes entrades (20, 25 i 35). Per poder discriminar entre aquests nodes fulla és necessari crear un nou node pare, que actuarà com a arrel. En el node pare s'insereix una còpia del valor corresponent a l'entrada inferior del node fulla de la dreta (20) i dos apuntadors cap als nodes fulla.



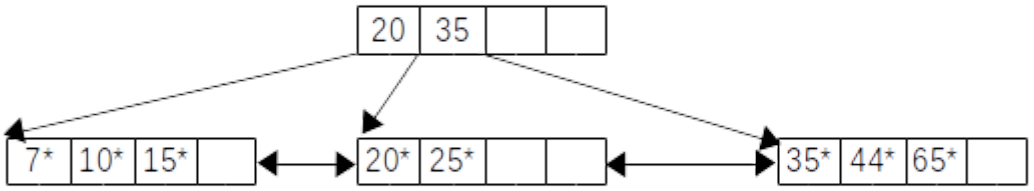
Inserim 44.
El node fulla al que els correspon situar el valor disposa d'espai lliure per a albergar la nova entrada, de manera que fem servir el node pare com discriminant per inserir cada entrada al node fulla corresponent. Després de la inserció es reordenen les entrades del node fulla.



Inserim 7.
Seguim el mateix raonament que en el cas anterior

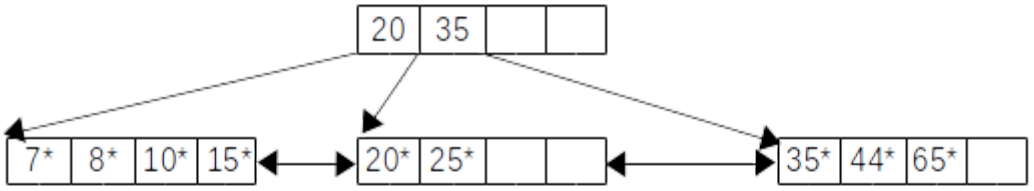


Inserim 65
L'entrada corresponent al valor 65 hauria inserir-se en el node fulla de la dreta. Com aquest node no té espai lliure, cal dividir-lo en 2. Dels dos nodes producte de la divisió, el primer albergarà els *d* primers valors (20 i 25) i el segon els *d+1* últims valors (35, 44 i 65). Per poder discriminar entre aquests dos nodes fulla cal inserir en el node pare el valor inferior del nou node fulla (35). El node pare disposa d'espai suficient per a albergar el nou valor, de manera que no es necessita cap reestructuració addicional de l'arbre.



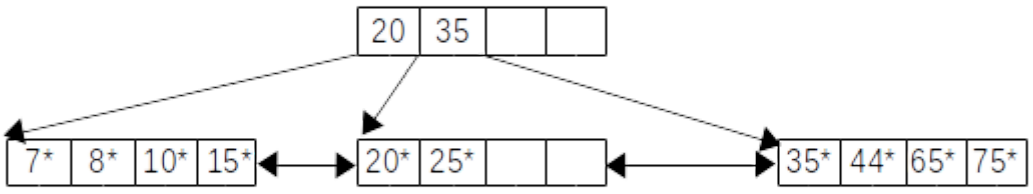
Inserim 8

A partir del node arrel es determina que l'entrada corresponent al valor 8 s'ha d'emmagatzemar en el primer node fulla. Com aquest node disposa d'espai lliure per a albergar noves entrades tan sols serà necessari reordenar les entrades del node després de la inserció.



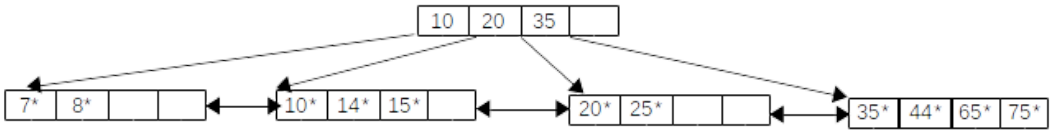
Inserim 75

A partir del node arrel es determina que l'entrada corresponent al valor 75 s'ha d'emmagatzemar en el tercer node fulla. Com aquest node disposa d'espai lliure per a albergar noves entrades tan sols serà necessari reordenar les entrades del node després de la inserció.



Inserim 14

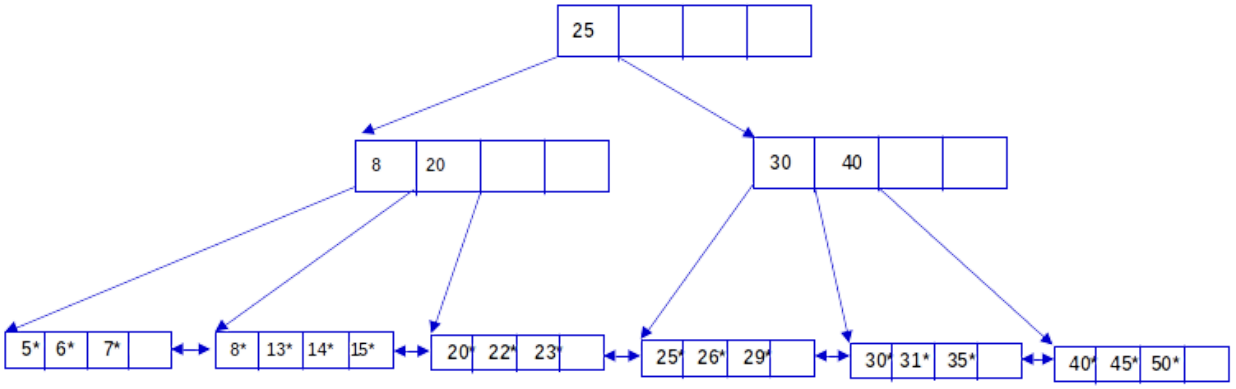
Li correspon la primera fulla, però no té espai lliure. Per a poder inserir la nova entrada cal dividir la fulla en dues.



Com aquest node no té espai lliure, cal dividir-lo en 2. Dels dos nodes producte de la divisió, el primer albergarà els *d* primers valors (7 i 8) i el segon els *d+1* últims valors (10,14 i 15). Per poder discriminar entre aquests dos nodes fulla cal inserir en el node pare el valor inferior del nou node fulla (10). El node pare disposa d'espai suficient per a albergar el nou valor, de manera que no es necessita cap reestructuració addicional de l'arbre.

Exercici 2 [25%]

Donat l'arbre B+ d'ordre 2 (d=2) següent :

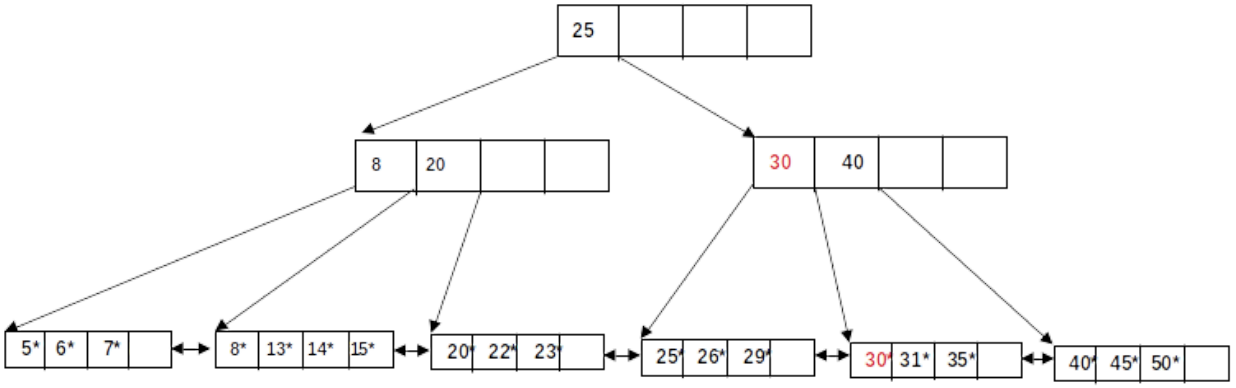


Mostra el resultat d'eliminar consecutivament els valors : **30-45-35-25**

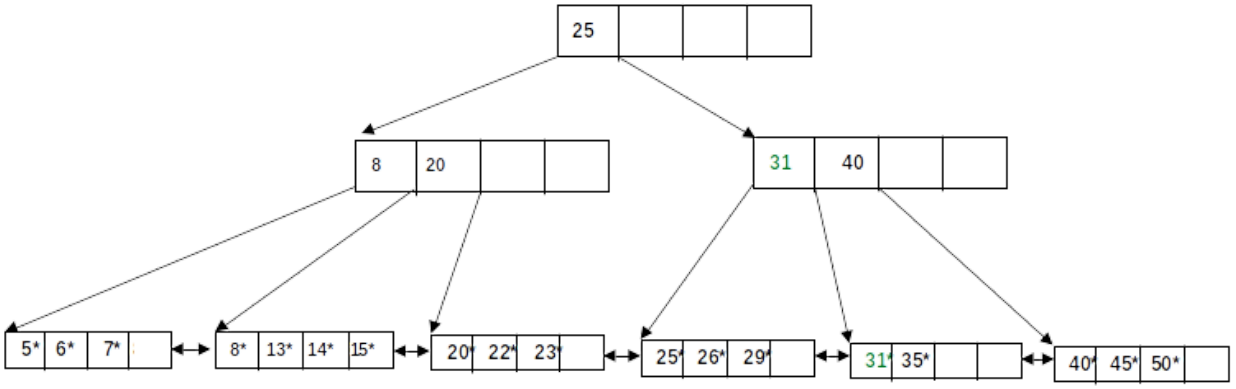
Cal especificar i argumentar les accions realitzades en cada operació i el resultat intermedi de l'arbre en cada pas.

Proposta de solució:

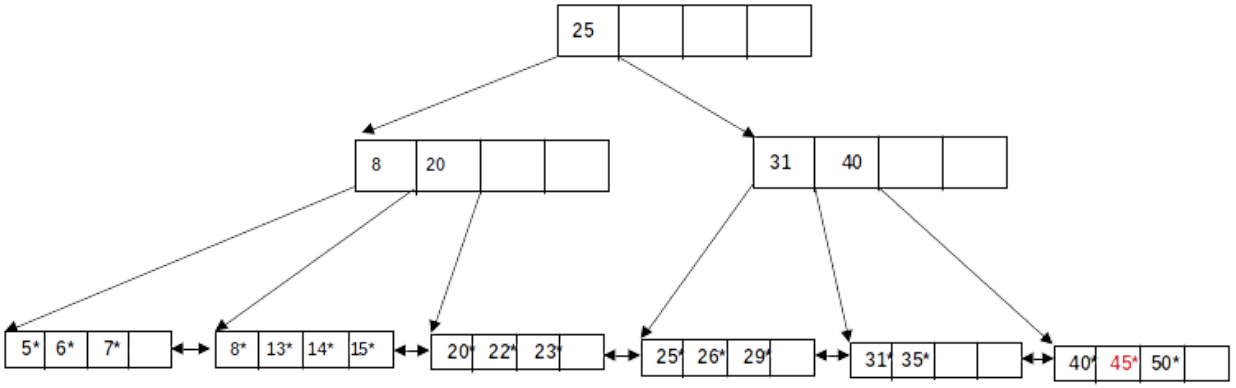
Eliminar el valor 30.
L'entrada corresponent al valor 30 es troba en el cinquè node fulla i té una còpia en el seu node pare. En conseqüència, prèviament a l'eliminació de l'entrada, es substituirà en el node pare la còpia del valor 30 pel valor corresponent a la següent entrada del node fulla (31). Posteriorment s'eliminarà l'entrada corresponent al valor 30.



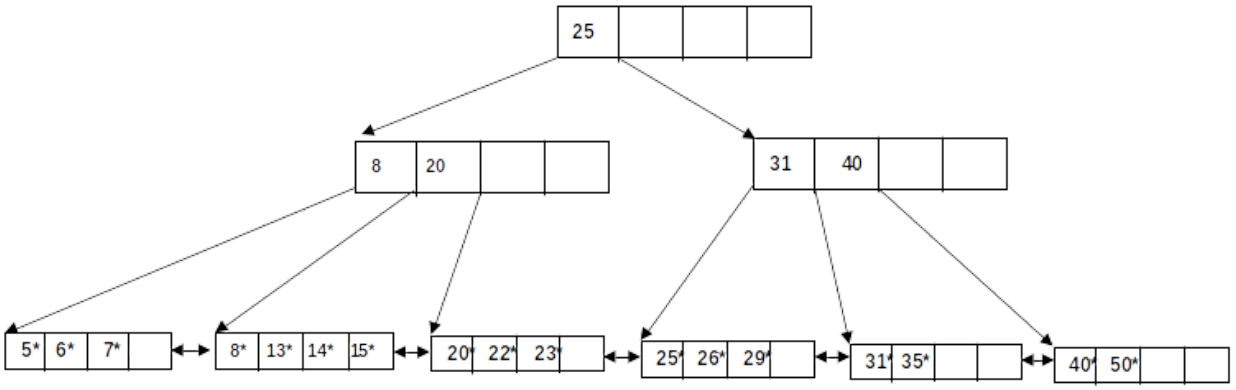
Després de l'eliminació, el node fulla afectat segueix tenint una ocupació igual o superior al 50%, per la qual cosa no serà necessària cap reestructuració addicional de l'arbre



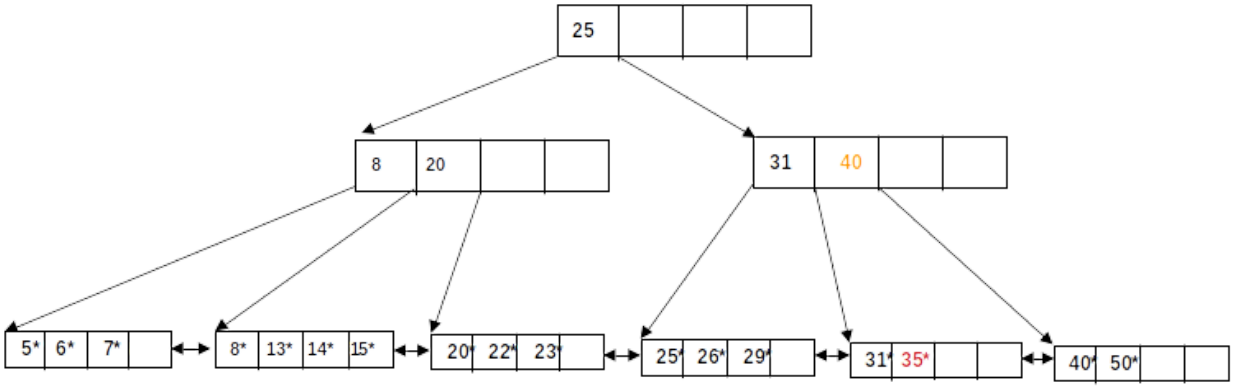
Eliminar del valor 45.
L'entrada corresponent al valor 45 es troba en l'últim node fulla.



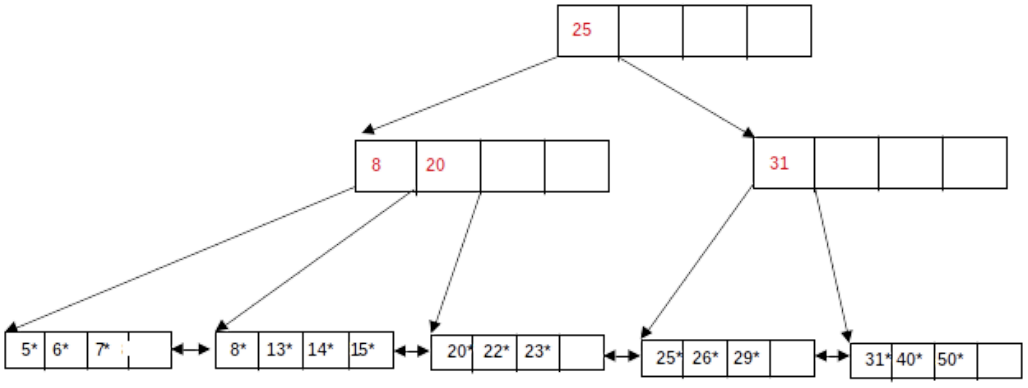
Després de l'eliminació, el node fulla afectat segueix tenint una ocupació igual al 50%, per la qual cosa no serà necessària cap reestructuració addicional de l'arbre. N'hi haurà prou amb reordenar les entrades del node fulla.



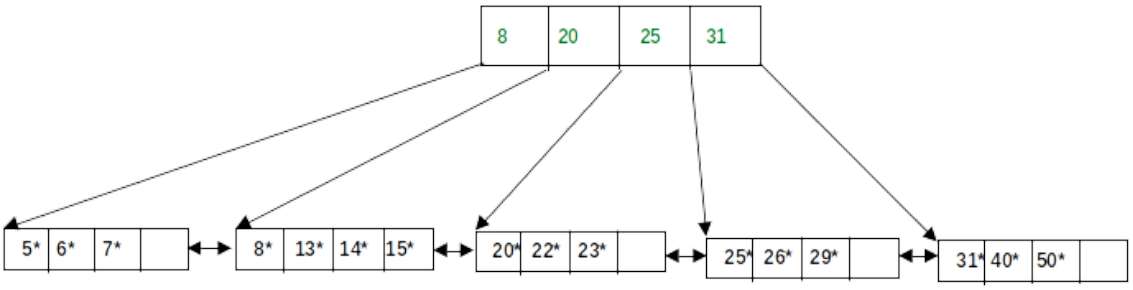
Eliminar del valor 35.
L'entrada corresponent al valor 35 es troba en el cinquè node fulla, l'ocupació és del 50%:



Després de l'eliminació, el node fulla afectat queda amb un nivell d'ocupació per sota del mínim permès d=2. El node germà de la seva dreta es troba en el nivell d'ocupació mínim, per la qual cosa no resulta factible fer una redistribució de les seves entrades. En conseqüència, tots dos nodes hauran fusionar-se i el node pare de tots dos ha de modificar-se per reflectir aquesta fusió. Després de la fusió, l'arbre perd el seu últim node fulla, les entrades passen al node germà de la seva esquerra i desapareix el valor 40 del node pare així com l'apuntador al node desaparegut.

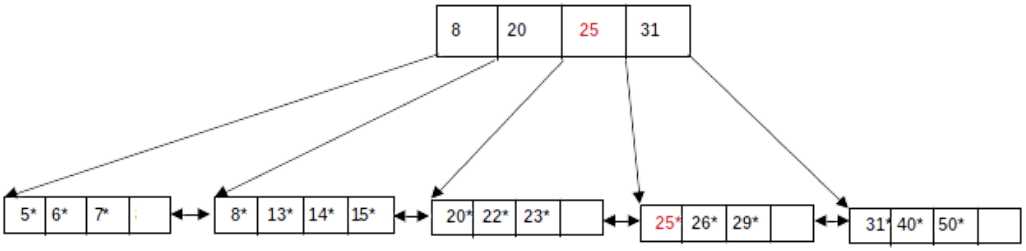


Ara, és el node pare dels nodes fusionats el que queda amb un nivell d'ocupació inferior al permès. Aquest node no té cap node germà a la seva dreta, per tant s'haurà de fusionar amb el node de l'esquerra.. Després d'aquesta nova fusió, l'arrel de l'arbre queda buida, per la qual cosa el node resultant d'aquesta última fusió es converteix en el node de nivell 1, i per tant en la nova arrel.

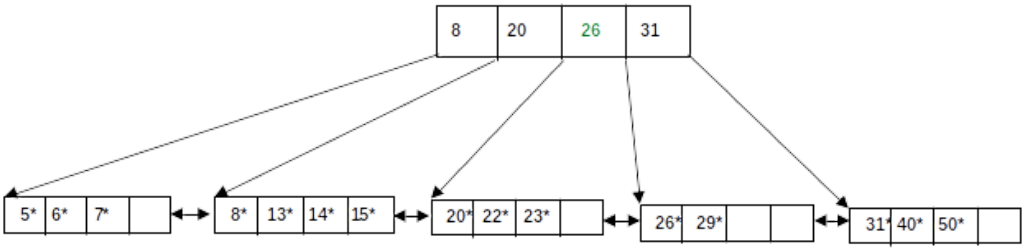


Eliminar del valor 25

L'entrada corresponent al valor 25 es troba en el quart node fulla, l'ocupació és superior al 50%, i té una còpia en el seu node pare. En conseqüència, prèviament a l'eliminació de l'entrada, es substituirà en el node pare la còpia del valor 25 pel valor corresponent a la següent entrada del node fulla (26). Posteriorment s'eliminarà l'entrada corresponent al valor 25.



Després de l'eliminació, el node fulla afectat segueix tenint una ocupació igual al mínim permès (d = 2), per la qual cosa no serà necessària cap reestructuració addicional de l'arbre i n'hi haurà prou amb la reordenació de les seves entrades.

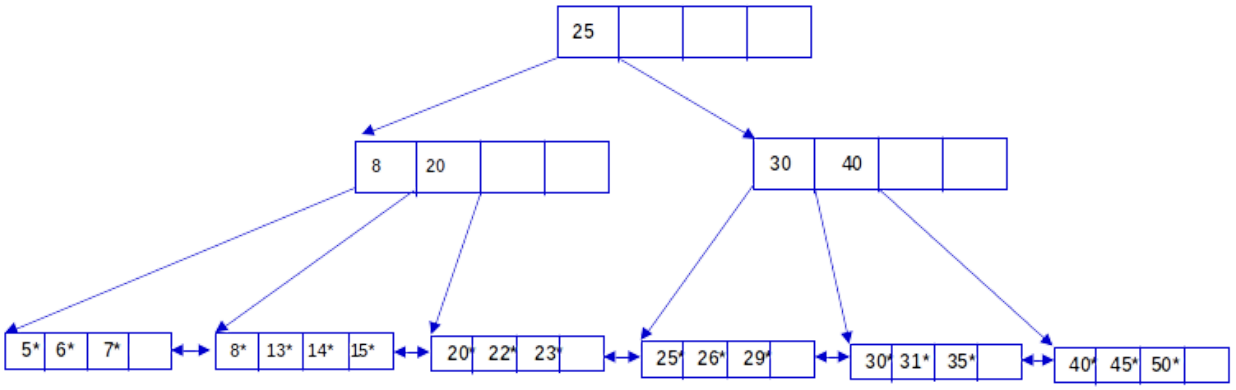


Exercici 3 [25%]

Donada la següent consulta:

```
SELECT first_name, last_name, job
FROM employee
WHERE id BETWEEN  22 AND 45;
```

i tenint en compte que el valor de `id` és el que està indexat a l'arbre B+ que es mostra a continuació.



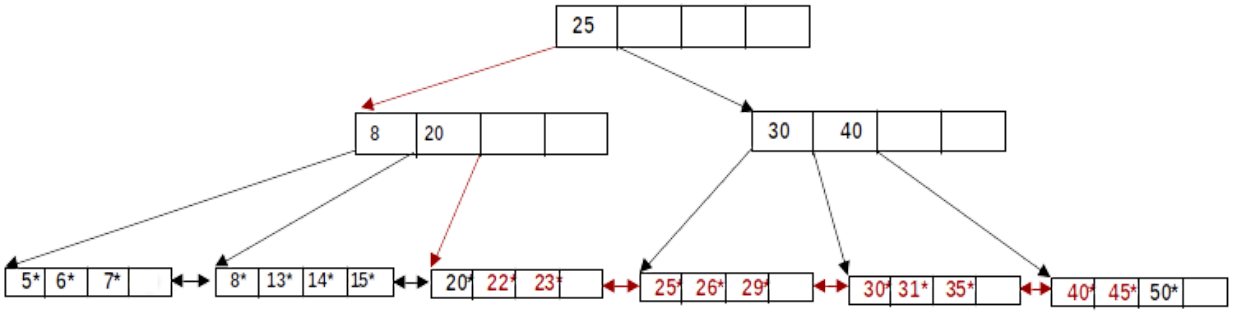
Es demana que indiquis els accessos que són necessaris per a obtenir les dades de les files resultants i, en segon lloc, per quines raons l'existència d'aquest arbre millora el rendiment de la consulta.

Proposta de solució:

Com s'explica en el mòdul 4 (implementació de mètodes d'accés) es realitza un accés seqüencial per valor:

- Busquem el primer valor 22:
- Com que 22<25 accedim a la branca de la esquerra
- Com que 22> 20 accedim a la fulla corresponent

- Localitzem el 22 i busquem el següents valors. En aquest node podem obtenir el 23 i passem a la següent fulla obtenint els valors 25,26 i 29.
- Continuem amb la següent fulla obtenint els valors 30, 31 i 35.
- Posteriorment accedim a la següent fulla i obtenim el valor 40 i 45
- En aquest moment aturem la cerca.



Cadascuna de les entrades recuperades de l'índex (22*, 23*, 25*, 26* **,** 29*, 30*, 31*, 35*,40* i 45*) estarà formada per un valor i un identificador de fila o RID.

Cal tenir en compte que per a cada valor obtingut podrem accedir a la resta de les dades de la consulta ja que estan indicades en el RID.

Mitjançant l'accés seqüencial per valor, un cop localitzada la primera entrada de l'índex poden obtenir ordenadament la resta d'entrades seguint l'apuntador existent en cada fulla de l'arbre cap a la següent fulla. Per tant, mitjançant l'ús d'índexs de tipus arbre B+ no cal localitzar en l'índex cada entrada del rang buscat de manera individual, amb el que es millora el rendiment de la consulta al reduir-se el nombre d'operacions E/S necessàries per recuperar el total d'entrades implicades en la consulta.

Exercici 4 [25%]

Exposa raonadament, amb precisió i detall els avantatges i inconvenients de l'ús dels índexs basats en arbres B+ i de l'ús dels índexs basats en dispersió. (límit de la resposta 1 pàgina)

Proposta de solució:

Els **índexs basats en arbres B+** serveixen per facilitar els accessos directe i seqüencial per valor d'un (o més) atributs. El seu objectiu primordial és intentar aconseguir que les cerques es facin amb un nombre reduït d'E/S.

Avantatges d'utilització d'arbres B+:

- Es poden fer cerques per accés directe i seqüencial per valor d'un (o més) atributs.
- Els arbres B+ permeten l'accés a les dades ordenat i les insercions i eliminacions es realitzen en temps logarítmic amortitzat. Només necessiten $\log_n N$ accessos per a realitzar operacions de cerca, inserció i eliminació, essent N el nombre de valors possibles.
- Són una eina molt eficient per a grans volums de dades.
- Per la seva estructura el temps de cerca és mínim.

Inconvenients d'utilització d'arbres B+:
Potser l'únic defecte que té un arbre B+ és que les pàgines poden estar utilitzades només al 50% de la seva capacitat. Per tant, això pot ser un malbaratament de memòria considerable si es té una quantitat de dades molt gran.
Crear i mantenir un arbre B+ és costós quan hi ha moltes operacions d'inserció/esborrat. Aquest també pot ser un factor que faci el seu ús inadequat.

Els **índexs basats en dispersió** faciliten l'accés directe per valor. La seva característica principal és que les entrades es col·loquen a les pàgines segons una funció de dispersió *h* (*p=h(v)*, on p=pàgina i v=valor).

Avantatges dels índexs basats en dispersió:

- Si estan ben dissenyats, poden aconseguir un rendiment una mica millor que els índexs arbre B+ en la implementació dels accessos directes per valor.
- Si l'entrada està en una pàgina primària només cal una E/S per localitzar una entrada.

Inconvenients dels índexs basats en dispersió:

- No serveixen per a l'accés seqüencial per valor.

- Per localitzar una entrada en pàgines excedents sempre caldrà fer més d'una E/S.
- Si el factor de càrrega és 0,5 el rendiment en temps serà excel·lent, però la meitat de l'espai es malbaratarà.
- Si la dispersió és estàtica i el nombre de dades indexades creix més del previst, pot passar que el factor de càrrega aumenti i el rendiment de l'índex empitjori.

Recursos

Els següents recursos són d'utilitat per la realització de la PAC:

Bàsics

- Mòdul didàctic 4. Implementació de mètodes d'accés.

Criteris de valoració

La ponderació dels exercicis és la següent:

- Exercici 1: 25%
- Exercici 2: 25%
- Exercici 3: 25%
- Exercici 4: 25%

Aquesta PAC s'ha de fer de manera estrictament individual. Qualsevol indici de còpia serà penalitzat amb un suspens (D) per a totes les parts implicades i la possible avaluació negativa de l'assignatura en la seva totalitat.

Format i data de lliurament

1. El format del fitxer ha de ser PDF.
2. El nom del fitxer ha de tenir el format següent: "nomUsuariUOC_PAC4.pdf", per exemple "jperezbr_PAC4.pdf".
3. El nom del alumne ha d'aparèixer a la portada i en cada pàgina del document.

Data límit de lliurament : 10 de maig de 2019 a les 24:00 hores.

La data de lliurament d'aquesta PAC ha de ser estrictament respectada, i no s'acceptarà cap lliurament després de la data establerta. Si es considera per alguna raó justificada que no es va a poder complir amb aquesta data, l'estudiant s'haurà de posar en contacte amb el seu consultor de l'assignatura amb suficient anterioritat per poder buscar conjuntament una solució al respecte. Si s'acorda el lliurament amb posterioritat, la nota màxima d'aquesta PAC serà un aprovat (C+).