Algoritmusok

Életből vett példák

* Tejforralás
* Villanykörte csere
* Pénzváltás
* Adott kerületű téglalapból maximális területű meghatározása
* Legolcsóbb repülőjegy megkeresése paraméterek alapján

Tejforralás:

Precondition:

* Van tejünk, több, mint amennyit forralni akarunk
* Van gáztűzhelyünk, van benne gáz
* A gáztűzhely alapállapotban van
* Képes felforralni a gáztűzhelyünk maximum fokozaton a tej mennyiségét, amit forralni akarunk

Postcondition:

* Van forralt tejünk
* Gáztűzhelyet hagyja az algoritmus alapállapotban

Kivétel:

* Odaégettük a tejet
* Ki tudjuk futtatni a tejet
* Közben elromolhat a gáztűzhely

1. Vegyünk egy legalább 25%-al nagyobb tűzálló edényt, mint amennyi tejet szeretnénk forralni
2. Öntsük bele a forralni kívánt tejet az edénybe
3. Az egyik gázrózsát gyújtsuk meg közepes lángon
4. Helyezzük az edényt a meggyújtott gázrózsára
5. Amennyiben 10 percen belül elkezd emelkedni a tej szintje, zárjuk le a gázrózsát, ha ez nem teljesül, állítsuk a fokozatot maximumra
6. Amennyiben elkezd emelkedni a tej szintje, zárjuk le a gázrózsát
7. Vegyük le az edényt a gázrózsáról

Bemenetek:  
Edény mérete, alakja, anyaga, tej mennyisége, tej összetétele, gázrózsa állapotai

Villanykörte csere fali lámpában

Precondition:

* Villanykörte ami nem működik és a foglalatban van
* A foglalat sérülésmentes
* Van áram a lakásban
* Van kapcsolónk, ami vezérli a körténk áramellátását
* Van otthon csere villanykörte

Postcondition:

* A villanykörte, ami a foglalatban van, működik

Kivétel:

* Áramütés

1. Kapcsoljuk le az áramot a villanykörtén a kapcsoló segítségével
2. Távolítsuk el a burkolatot a lámpatesten akkora mértékben, hogy a cserét végre tudjuk hajtani és a burkolatot sérülésmentesen vissza tudjuk helyezni
3. Távolítsuk el a foglalatból a nem működő villanykörtét a foglalat vonatkozó dokumentációja alapján
4. A csere villanykörte csomagolását távolítsuk
5. A csere villanykörtét helyezzük a foglalatba a csere villanykörte vonatkozó dokumentációja alapján
6. Engedjük el a behelyezett villanykörtét
7. Helyezzük áram alá a foglalatot a kapcsoló segítségével
8. Ha világít a körte, lépjünk tovább
9. Ha nem világít a körte, keressünk egy másik, működő foglalatot, ami az aljzatnak megfelelő és teszteljük, hogy a csere villanykörténk működik e
10. Ha nem működik a körte, menjünk a 3. pontra
11. Helyezzük vissza a burkolatot a lámpatestre
12. A nem működő villanykörtéket helyezzük a hulladékba a vonatkozó dokumentációja alapján

Adathalmazok

* Adatbázisban keresés
* Adatbázisban eltárolás
* Rendezetlen halmaz rendezése - házi feladat

Adatbázisban keresés

Precondition:

* indexált adatbázisunk
* vannak benne nem üres értékek
* meg van adva, hogy milyen adatot szeretnénk kinyerni az adatbázisból

Postcondition:

* Rendelkezésre áll az információ, amit ki akartunk nyerni

Attribútummal rendelkező sor létezésének lekérdezése

1. Olvassuk be, hogy milyen attribútum alapján szeretnénk keresni, tároljuk el ezt A-ban
2. Olvassuk be, hogy milyen értéket szeretnénk keresni, jelöljük ezt V-vel
3. Tároljuk el az A-nak megfelelő oszlop első elemét egy listába, jelöljük L-el
4. Tároljuk el az A-nak megfelelő oszlop következő elemét egy listába, jelöljük L-el
5. Amennyiben van a következő sorban index érték, lépjünk a 4-es pontba
6. Hajtsunk végre egy keresést az L-ben, V-t keressük
7. Amennyiben a keresés azt mondta, hogy van, adjunk vissza IGAZ-t
8. Adjunk vissza HAMIS-t

Adjuk vissza, hogy milyen attribútumok vannak az adatbázisunkban

1. Beolvassuk a bal felső sarokban lévő értéket egy L listába
2. Beolvassuk az eggyel jobbra lévő értéket
3. Amennyiben az eggyel jobbra lévő érték üres, adjuk vissza az L-t
4. Lépjünk a 2-es lépésre

Kimenetek:

* milyen attribútumok vannak az adatbázisban
* van e attribútum ami nincs kitöltve
* van e olyan attribútum, ami több sorban is előfordul
* van e olyan sor, aminek a név attribútuma xy
* vannak e benne teljesen megegyező sorok (indexen kívül)
* vannak e benne olyan sorok, amik csak az életkorban különböznek (indexen kívül)
* adja vissza azokat a sorokat, amiknek az életkorra valamilyen logikai kifejezést teljesít
* adja vissza azokat a sorokat, amiknek valamelyik attribútuma valamilyen logikai kifejezést teljesít
* például a felső kettő egy kombinációját
* adja vissza, hogy pl az 1 indexű sornak mi a név attribútuma
* adja vissza egy halmazt azokból az attribútumokból, amiknek a sora valamilyen logikai kifejezést teljesít
* igen-nem
* magáról a adatbázisról kérdezünk - pl hány entry, van e index, milyen attribútumok vannak benne
* adjon nekünk vissza sorokat
* adjon nekünk vissza attribútumokat
* használhatunk logikai kifejezéseket

Adatbázisban eltárolás

Precondition

* Van egy nem üres indexelt adatbázisunk
* Van egy sorunk, amit hozzá akarunk adni
* Sorban tudunk menni az új sor attribútumain és el tudjuk döntei, hogy van e még olyan, aminek van értéke

Postcondition

* A sorunk szerepel az adatbázisban
* Nem hoztunk létre duplikációt

Kivétel

* Tele van az adatbázis

1. Megkérdezzük az adatbázist, hogy van e hely új sornak, ha nincs, akkor kilépünk egy hibával
2. Megkérdezzük az adatbázist, hogy van e a sorral megegyező sor már benne, ha igen, akkor végeztünk
3. Megkeressük, hogy melyik a legnagyobb index, jelöljük I-vel
4. Létrehozunk egy új sort, aminek az indexe I+1 lesz
5. Ha van következő értékkel rendelkező attribútum, beírjuk a sor megfelelő cellájába, egyébként végeztünk
6. Menjünk vissza a 4-as pontba

Kicsit technikaibb

* Alakzat rajzolása
* File-ok csoportosítása könyvtárban
* Szavak megszámlálása szövegben

Csomó file különböző mappákban egy meghajtón.

Minden mappában minden file-t feldolgozunk.

Legyenek mappákba rendezve kategóriák szerint - hang, szöveg, videó.

Mappákon belül pedig kiterjesztések szerint legyenek rendezve.

Megegyező nevű file-ok esetén adjunk sorszámot a file-nak ami ütközik egy másikkal (file név végére)

Precondition:

* Van írási és olvasási jogunk a file-okhoz és a mappákhoz
* Senki más nem fogja éppen a file-okat
* Van listánk arról, hogy mi micsoda

Postcondition:

* Legyenek mappákba rendezve kategóriák szerint - hang, szöveg, videó.
* Mappákon belül pedig kiterjesztések szerint legyenek rendezve.

1. Vesszük a következő elemét a kiterjesztés listánknak, ezt jelüljük K-val, ha nincs, végeztünk
2. Ellenőrizzük, hogy van e a megfelelő kategóriának mappája, ha nincs létrehozzuk
3. Létrehozunk egy mappát K néven
4. Mélységi kereséssel készítünk egy listát a mappákból M néven
5. Ha M-nek van új eleme, az új elemet megvizsgáljuk, kilistázzuk az összes file-t a mappában, egy L listába, ha nincs, megyünk az 1-es pontba
6. Amíg L-nek van új eleme, az új elemet megvizsgáljuk, hogy K e a kiterjesztése, ha igen, megvizsgáljuk, hogy van e már a K mappában egyező nevű elem, ha igen, menjünk a 7-es pontba, ha nem, menjünk a 9-as pontba
7. Konkatenálunk a file nevéhez egy 1-est
8. Megvizsgáljuk, van e a mappában egyező nevű elem, ha van, lépünk a 7-es pontba
9. Átmásoljuk a file-t a K mappába
10. Menjünk vissza a 5-ös pontba

Illeszkedjünk valamihez

* Levél feladás
* Hajtsunk át egy körforgalmon
* Vegyünk egy kiló kenyeret

Programozás felé

* Tároljunk el valamit excelben, ha még nincs eltárolva
* Ugyanez SQL-ben
* Létesítsünk egy IP címen és egy porton keresztül egy socketet