Структури от данни

Специалност Информационни системи

Задачи за домашна работа №1

Декември, 2018

Задача 1

Напишете програма, която приема като аргумент от командния ред две имена на текстови файлове. Първият съдържа “компресиран” низ, а във втория вие трябва да запишете декомпресираната му версия.

Компресираният низ съдържа 3 вида конструкции:

- Символи (всякакви символи, различни от скоби, цифри и наклонена черта ‘\’)

- Екранирани символи - наклонена черта, следвана от произволен символ. Така в компресираният низ може да се сложат специални символи

- Групи, които започват с число, последвано от скоби, съдържащи компресиран низ. Декомпресират се като се декомпресира низа в скобите и се повтори толкова пъти колкото е числото.

Примери:
AABC -> AABC
A\\BC -> A\BC
R2(AB)3(Z) -> RABABZZZ
R2(A\2B)3(Z) -> RA2BA2BZZZ
AB12(X)2(B3(A)) -> ABXXXXXXXXXXXXBAAABAAA

Не се позволява използването на рекурсия, както и структури от данни или алгоритми от стандартната библиотека.

Задача 2

Реализирайте СД Двоично наредено дърво, като да поддържа следните операции:

- Добавяне на елемент
- Изтриване на елемент
- Търсене на елемент

Реализацията да поддържа произволен тип данни, за който е дефиниран оператор за сравнение.

Реализацията да се използва за реализиране на следната програма:

- Като параметър от командния ред на програмата се подава име на двоичен файл, за който всеки запис се състои от три полета:
  - Цяло число key (32 бита), последвано от
  - Размера на данните в байтове (32 бита) и
  - Символен низ data (не-терминиран)ч

  Да се построи идеално балансирано двоично дърво съдържащо записаните в него данни.

- Програмата да поддържа (чрез диалогов режим) следните четири заявки. Не се изисква дървото да остане идеално балансирано след изпълнението на заявките.
  - *add <key> <data>* - добавя в дървото елемент с посочените данни (низът data е до края на реда).
  - *remove <key> <data>* - извежда на стандартния изход true/false в зависимост дали елемент с дадените данни е намерен и премахнат.
  - *removeall <key>* - премахва от дървото всички елементи с дадения key и извежда на стандартния изход броя на премахнатите елементи.
  - *search <key>* - извежда на стандартния изход данните, съответни на дадения ключ, или съобщение за грешка, ако няма такъв ключ.

- Програма приключва изпълнението си при достигне до символа за край на файл в стандартния вход (представян с ctrl+z или ctrl+d при четене от клавиатура).

Задача 3

Изпращането на дървета за коледа се превръща в истински проблем за пощенските
служители. Всеки иска да изпрати своето дърво на приятел. След обстоен анализ на
пратките, служителите забелязали, че голяма част от дърветата, които се изпращат, имат
еднаква структура и се различават само по стойностите на елементите във възлите
си. Това веднага ги навело на мисълта, че значително ще намалят
работата си, ако могат да разберат кога две дървета имат еднаква структура (или,
както се изразявали математиците, за да звучат по-важно, кога са изоморфни).

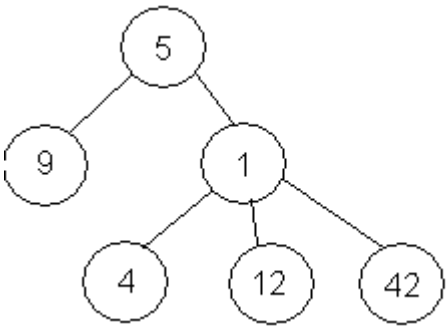
Тъй като служителите си имат друга работа, а и не разбират много от тези неща,
решили като да бъдат добри хора (за да заслужат подаръци все пак) и да прехвърлят
камъка във вашата градина. Задачата ви е да напишете програма, която по зададени две дървета решава дали те са изоморфни.

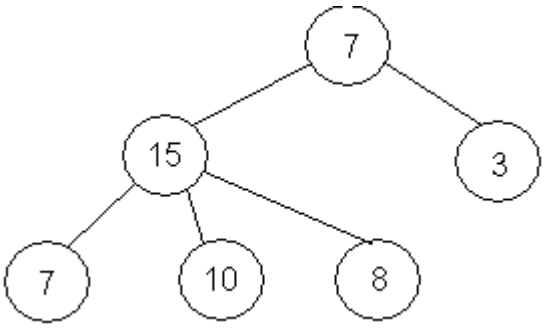
Точната дефиниция на изоморфни дървета е:

- Празните дървета са изоморфни
- Дървото T с k поддървета $t_1,..,t_k$ е изоморфно на дървото Q с k поддървета $q_1,..,q_k$, ако има такава биекция $b:[1,k] \rightarrow [1,k]$, че за всяко $i \in [1,k]$ t_i е изоморфно на $q_{b(i)}$.

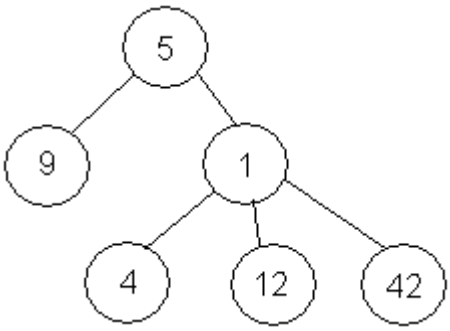
С други думи, игнорираме стойностите във възлите и подредбата на под-дърветата, ако има такава.

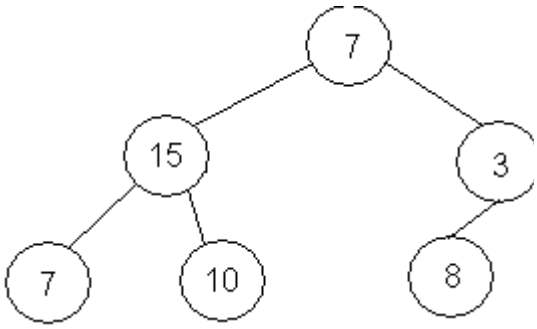
Например, следните дървета са изоморфни:





Но пък следните две не са:





Дърветата представяме като символни низове по следния начин:

- Празното дърво се представя с низа “{}”
- Дървото T със стойност на корена X и представяне на поддърветата му като низовете S_1,..,S_k се представя с низа “(“ ++ X ++ “{“ ++ S_1 ++ .. + S_k ++ “})”.

Съответното представяне на първото дърво от фигурата е:
(5 {(9 {}) (1 {(4 {}) (12 {}) (42 {})})})
А на последното:
(7{(15 {(7 {}) (10 {})}) (3 {})})

Напишете програма, която прочита от командния ред два символни низа, описващи две дървета и извежда на екрана YES или NO в зависимост дали съответните дървета са изоморфни или не.

Задача 4

Играта разполага с игрално поле от 2N+1 полета, като в началото в най-десните N полета имаме разположени жаби, гледащи наляво, а в най-левите полета - жаби, гледащи надясно. Целта на играта е жабите да си разменят местата и да се стигне до противоположната конфигурация:

Правилата за игра са следните: всяка жаба може да се движи само в посоката, която гледа. Всяка жаба може да скочи на свободно място пред себе си или да прескочи **една** жаба, за да отиде на празно място пред нея.

Вход: N – броят на жабите гледащи в една посока

Изход: всички конфигурации, през които се минава за да се стигне от началното до финалното състояние.

Примерен вход: 2

Примерен Изход:

>>_<<

>_><<

><>_<

><><_

><_<>

_<><>

<_><>

<<>_>

<<_>>

Подсказка: Помислете какви начини на обхождане можем да приложим.

Задача 5

Даден е текстов файл, който се състои от поредица от думи, които се състоят само от малките латински букви. Да се напише програма, която създава речник, състоящ се от ключове - думите от подадения документ, и асоциирани стойности - поредният номер, указващ мястото на думата във файла. Да се реализират методи за добавяне на дума в речника, за премахване на дума от него, както и за намиране на стойност по ключ - подадена дума. Да се реализират и всички други необходими методи за динамичните структури от данни.

Пример:
Нека във файл с име example.txt се съдържат думите bow dog bank
Създаденият речник трябва да има следния вид: