**Imię i nazwisko:** Jakub Pranica **Grupa:** Informatyka -13i **Data:** 06-03-2017 **Przedmiot:** Metody Programowania

**Wykładowca:** dr inż. Zbigniew Kokosiński **Semestr:** 2

**Temat:** Napisać algorytm generujący wszystkie możliwe sekwencje drzewa t-arnego i wypisać ich reprezentacje typu X na ekran lub do pliku.

**Drzewem decyzyjnym** nazywamy drzewo, w którym każdy wewnętrzny wierzchołek reprezentuje pewną decyzję zaś liście zawierają wszystkie możliwe wyniki.

**Drzewem t-arnym** nazywamy strukturę zdefiniowaną rekurencyjnie na skończonym zbiorze węzłów, która nie zawiera żadnych węzłów albo składa się z t+1 rozłącznych zbiorów węzłów: korzenia, oraz t różnych poddrzew t-arnych.

**Opis programu:** Program prosi użytkownika o wprowadzenie liczby n oraz liczby t. Gdzie n oznacza ilość rozgałęzień (węzłów) natomiast t ilość gałęzi w jednym rozgałęzieniu. Następnie generuje zbiór {1,1+t,1+2t,...} i zapisuje go w tablicy - a. Czyli mamy a[0]=1, natomiast każdy kolejny wyraz jest większy o t od poprzedniego: a[i] = a[i-1]+t. Tablica a będzie zbiorem reprezentacji typu Z kolejnych możliwych drzew t-arnych. Po inicjalizacji następuje transformacja reprezentacji Z na reprezentację typu X - wynik zostaje wypisany na ekran lub do pliku. Po wypisaniu tablica a przekształcana jest tak aby reprezentowała kolejną sekwencję drzewa t-arnego w zapisie Z. Program wykonuje tę czynność dopóki nie uzyska wszystkich sekwencji drzewa t-arnego. Na koniec podaje dodatkowo ich ilość. **(Pseudokod na odwrocie kartki)**

**Napotkane problemy: 1.**Reprezentacja zbiorów, za pomocą tablic w języku programowania c++ **2.**Wygenerowanie wszystkich wartości funkcji wyboru (wszystkich możliwych sekwencji drzewa t-arnego o podanych przez użytkownika właściwościach) **3.** Transformacja wygenerowanej reprezentacji Z na reprezentację typu X drzewa t-arnego **4.**Wypisanie na ekran i do pliku reprezentacji Z oraz X zapisanej w postaci tablicy.

**Przykładowy wynik programu:**

Program generuje wszystkie możliwe reprezentacje typu Z (w kolejnosci RevLex) drzewa t-narnego o podanych właściwościach i zamienia go na zapis w reprezentacji typu X. Gdzie wypisać wyniki? 1 - na ekran 2 - do pliku 1

Podaj n: **3** Podaj t: **4**

Rozpoczynam wypisywanie kolejnych możliwych drzew t-arnych o podanych danych w reprezentacji X (dodatkowo w nawiasie podaje zapis w reprezentacji typu Z)

1. x=100010001000 z:1 5 9

2. x=100010010000 z:1 5 8

3. x=100010100000 z:1 5 7

4. x=100011000000 z:1 5 6

5. x=100100001000 z:1 4 9

6. x=100100010000 z:1 4 8

7. x=100100100000 z:1 4 7

8. x=100101000000 z:1 4 6

9. x=100110000000 z:1 4 5

10. x=101000001000 z:1 3 9

11. x=101000010000 z:1 3 8

12. x=101000100000 z:1 3 7

13. x=101001000000 z:1 3 6

14. x=101010000000 z:1 3 5

15. x=101100000000 z:1 3 4

16. x=110000001000 z:1 2 9

17. x=110000010000 z:1 2 8

18. x=110000100000 z:1 2 7

19. x=110001000000 z:1 2 6

20. x=110010000000 z:1 2 5

21. x=110100000000 z:1 2 4

22. x=111000000000 z:1 2 3

Ilosc wygenerowanych drzew: 22 Ilosc lisci: 10

Podaj n: **1** Podaj t: **10**

1. x=1000000000 z:1

Ilosc wygenerowanych drzew: 1 Ilosc lisci: 10

Podaj n: **10** Podaj t: **1**

1. x=1111111111 z:1 2 3 4 5 6 7 8 9 10

Ilosc wygenerowanych drzew: 1 Ilosc lisci: 1

Podaj n: **4** Podaj t: **2**

1. x=10101010 z:1 3 5 7

2. x=10101100 z:1 3 5 6

3. x=10110010 z:1 3 4 7

4. x=10110100 z:1 3 4 6

5. x=10111000 z:1 3 4 5

6. x=11001010 z:1 2 5 7

7. x=11001100 z:1 2 5 6

8. x=11010010 z:1 2 4 7

9. x=11010100 z:1 2 4 6

10. x=11011000 z:1 2 4 5

11. x=11100010 z:1 2 3 7

12. x=11100100 z:1 2 3 6

13. x=11101000 z:1 2 3 5

14. x=11110000 z:1 2 3 4

Ilosc wygenerowanych drzew: 14 Ilosc lisci: 5

**Pseudokod:**

**Wypisz na ekran:** "Program generuje wszystkie możliwe reprezentacje typu Z (w kolejnosci RevLex) drzewa t-narnego o podanych właściwościach i zamienia go na zapis w reprezentacji typu X. \n"

**Użytkownik podaje liczbę**, a program przypisuje ją do zmiennej **n; //**ilość rozgałęzień

**Użytkownik podaje liczbę**, a program przypisuje ją do zmiennej **t; //**ilość gałęzi w jednym rozgałęzieniu

**Zdefiniuj tablicę typu int a i wypełnij ją kolejnymi elementami zbioru:**

a = {1,1+t,1+2t,...} //Tablica a zawiera zbiór, który jest reprezentacją typu Z kolejnych możliwych drzew t-arnych.

**Wypisz na ekran:** "Rozpoczynam wypisywanie kolejnych możliwych drzew t-arnych o podanych danych w reprezentacji X (dodatkowo po spacji podaje zapis w reprezentacji typu Z) \n "

**Zdefiniuj** int ilosc = 0;

**Powtarzaj:**

**Zwiększ** ilosc **o 1 (**ilosc++;**)**

**Uruchom funkcję** wypiszXdrzewa(n, t, a); //funkcja zamienia reprezentację typu Z zawartą w tablicy a na reprezentację typu X i wypisuje ją na ekran lub do pliku

**Wypisz na ekran lub do pliku:** " z: \***tu zawartość tablicy a**\* ) \n" //tablica a zawiera reprezentację drzewa w zapisie typu Z

**Ustaw tablicę a tak aby przedstawiała kolejną wartość funkcji wyboru (kolejne drzewo w zapisie Z):**

for(int i = 1; n-i-1>=0 ; ){

if( a[n-i] == a[n-i-1] ){

a[n-i-1]--;

a[n-i] = t\*(n-i)+1;

i++;

} else break;

}

**dopóki** a[0] == 1; //dopóki nie wygenerujesz wszystkich nieporządków

**Wypisz na ekran lub do pliku:** "Ilosc wygenerowanych drzew: " ilosc**;**

**Wypisz na ekran lub do pliku:** " Ilosc lisci: " (t-1)\*n+1**;**

**Funkcja** wypiszXdrzewa(n, t, a):

**Wypisz na ekran:** ilosc + ". x="

**Zdefiniuj** int i = 1; oraz int j = 0;

**Powtarzaj:**

**Jeżeli** j<n && a[j]==i **jest prawdziwe to:**

**Wypisz na ekran:** "1"

**Zwiększ** j **o 1 (**j++;**)**

**w przeciwnym wypadku:**

**Wypisz na ekran:** "0

**koniec warunku.**

**Zwiększ** i **o 1 (**i++;**)**

**dopóki** i <= t\*n**;** // t\*n oznacza długość X'a

**Pełny kod programu w C++ dostępny pod adresem:** http://pastebin.com/MctRU5Yr

**Wnioski:** Program działa i prawidłowo generuje wszystkie możliwe sekwencje drzewa t-arnego dla podanych przez użytkownika danych takich jak: Ilość rozgałęzień (n węzłów) oraz ilość gałęzi w jednym rozgałęzieniu (t). Czas wygenerowania oraz wypisania wszystkich sekwencji drastycznie rośnie wraz ze wzrostem wartości n i/lub wartości t. Program pozwolił na uzyskanie następujących wyników: B(3,4)=22 B(1,10)=1 B(10,1)=1 B(4,2)=14 Regularnym drzewem t-arnym nazywamy drzewo t-arne, w którym każdy węzeł jest albo liściem albo ma stopień t. Liczba liści w regularnym drzewie t-arnym o n węzłach wewnętrznych jest równa (t-1)n + 1. Program zatem podaje dodatkowo prawidłową ilość liści dla danego drzewa.

**Literatura:**

* pl.wikipedia.org/wiki/Drzewo\_(informatyka) oraz pl.wikipedia.org/wiki/B-drzewo
* "Analiza Algorytmów - Ilościowa i jakościowa analiza drzew. Drzewa decyzyjne." - Aleksandra Orpel
* "Kombinatoryka dla programistów" - Lipski Witold