

- **Opis problemu:**

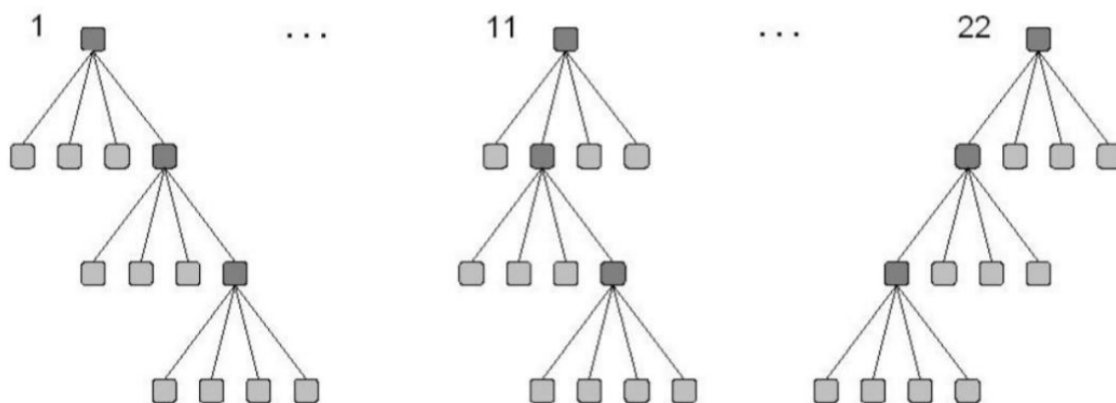
**Struktura drzewa:**

Drzewo (ang. tree) - jest strukturą danych zbudowaną z elementów, które nazywamy węzłami (ang. node). Dane przechowuje się w węzłach drzewa. Węzły są ze sobą powiązane w sposób hierarchiczny za pomocą krawędzi (ang. edge), które zwykle przedstawia się za pomocą strzałki określającej hierarchię. Pierwszy węzeł drzewa nazywa się korzeniem (ang. root node).

Liczbę drzew  $t$ -narnych można przedstawić za pomocą liczby **Catalana**:

$$B(n, t) = \frac{(t * n)!}{n! (t * n - n + 1)!}$$

( $A$ -porządek,  $n=3$ ,  $t=4$ , liczba drzew  $B(3,4)=22$ )



W informatyce szczególnie ważnym przypadkiem jest drzewo w którym liczba synów każdego wierzchołka wynosi nie więcej niż dwa. t.z. **drzewo binarne**. Często używa się go do przyspieszania wyszukiwania, operowania na posortowanych danych czy ich reprezentacji.

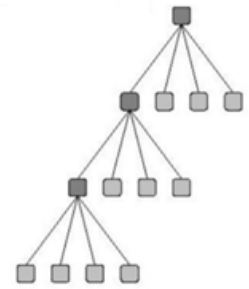
Znaczenie tych struktur jest bardzo duże i ze względu na swoje własności drzewa są stosowane praktycznie w każdej dziedzinie informatyki (np. bazy danych, grafika komputerowa, przetwarzanie tekstu, telekomunikacja, serwery).

- **Opis algorytmu:**

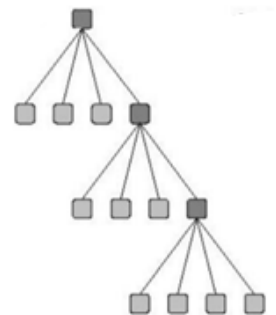
Algorytm generuje wszystkie drzewa  $t$ -narne reprezentując je binarnie w malejącym porządku leksykograficznym. W tym celu operuje na dynamicznej tablicy przesuwając w odpowiedni sposób jedynki

### Generowanie drzew t-narnych:

- o Tworzę dynamiczną tablicę o rozmiarze  $n \cdot t$  i wypełniam ją zerami,
- o na pierwsze  $n$  pozycji tablicy wstawiam 1, generując w ten sposób ostatnie drzewo np.:  
111000000000
- o wyszukuję pierwszą jedynekę zaczynając od prawej,
- o jeżeli pozycja znalezionej jedynki jest mniejsza niż  $n \cdot t - t$  to przesuwam ją na prawo, uzyskując w ten sposób jedną z generowanych sekwencji,  
110000001000
- o jeżeli pozycja znalezionej jedynki jest równa  $n \cdot t - t$  oznacza to że, należy przesunąć następną jedynekę np.:  
101100000000
- o algorytm wykonuje się rekurencyjnie do momentu gdy wszystkie jedynki znajdą się na wyznaczonych miejscach (w równych odstępach od siebie) np.:  
100010001000



Pierwsze wygenerowane drzewo w malejącym porządku leksykograficznym



Ostatnie wygenerowane drzewo

### • Fragment kodu:

Funkcja wykorzystująca rekurencję:

```
void tree(int p=n*t-1)
{
    p = search(p);
    if(p != n*t-t)
    {
        show();
        tab[p] = 0;
        tab[p+1] = 1;
        tree();
    }
    else if(p == n*t-t)
    {
        show();
        if(check(p));
        else
        {
            tab[p] = 0;
            p = search(p);
            tab[p] = 0;
            tab[p+1] = 1;
            tab[p+2] = 1;
            tree();
        }
    }
}
```

## Przykładowe wywołanie programu:

Generowanie drzew t-narnych z n wewnętrznymi węzłami.

Podaj liczbę t: 4

Podaj liczbę n: 3

```
111000000000
110100000000
110010000000
110001000000
110000100000
110000010000
110000001000
101100000000
101010000000
101001000000
101000100000
101000010000
101000001000
100110000000
100101000000
100100100000
100100010000
100100001000
100011000000
100010100000
100010010000
100010001000
```

Generowanie drzew t-narnych z n wewnętrznymi węzłami.

Podaj liczbę t: 3

Podaj liczbę n: 3

```
111000000
110100000
110010000
110001000
110000100
101100000
101010000
101001000
101000100
100110000
100101000
100100100
```

- **Wnioski:**

Programy poprawnie generuje drzew t-narne z n wewnętrznymi węzłami i wyświetla ich reprezentacje binarną w konsoli Windows. Jest on odporny na błędy użytkownika. Spełnia wszystkie założone kryteria.

Źródła:

- [1]. Kokosiński Z.: Parallel enumeration of t-ary trees in ASC SIMD model, Int. J. Computer Science and Network Security, Vol. 11, No. 12, 38-49 (2011),
- [2]. Ron Rivest i Thomas H. Cormen.: 'Wprowadzenie do algorytmów',
- [3]. Drzewa w informatyce.: [pl.wikipedia.org/wiki/Drzewo\\_\(informatyka\)](http://pl.wikipedia.org/wiki/Drzewo_(informatyka))