Matematyka dyskretna L, Lista 8 - Tomasz Woszczyński

Zadanie 1

Niech $A(x)$ będzie funkcją tworzącą ciągu a_n. Podaj postać funkcji tworzącej dla ciągu $s_n = a_0 + a_1 + a_2 + \ldots + a_n$. *Wskazówka*: Trzeba użyć funkcji tworzącej $\frac{1}{1-x}$.

Skoro $s_n = a_0+a_1+a_2+\ldots+a_n = \sum_{k=0}^{n} a_k$, to w łatwy sposób możemy podstawić s_n w postaci sumy do wzoru na funkcję tworzącą, a następnie całe wyrażenie przekształcić do takiej postaci:

$$\begin{aligned} A(x) &= \sum_{n=0}^{\infty} s_n x^n = \sum_{n=0}^{\infty} \left(\sum_{k=0}^{n} a_k \right) x^n = \sum_{n=0}^{\infty} \left(\sum_{k=0}^{n} \left(a_k x^k \right) \cdot x^{n-k} \right) \\ &= \left(\sum_{n=0}^{\infty} a_n x^n \right) \cdot \left(\sum_{n=0}^{\infty} 1 \cdot x^n \right) = A(x) \cdot \frac{1}{1-x} = \frac{A(x)}{1-x} \end{aligned}$$

Zadanie 2

Wyznacz funkcje tworzące poniższych ciągów. Wszędzie przyda się funkcja tworząca $\frac{1}{1-x}$, a w ostatnim podpunkcie będzie to odpowiednia potęga tej funkcji:

1. $a_n = n^2$
2. $a_n = n^3$
3. $\binom{n+k}{k}$

Przykład 1: Chcemy uzyskać funkcję tworzącą ciągu $\langle n^2 \rangle = \langle 0, 1, 4, 9, 16, \ldots \rangle$. Niech A przedstawia funkcję tworzącą $0 + 1x + 4x^2 + 9x^3 + \ldots$, weźmy wtedy funkcję $-xA$, którą przedstawia następujący szereg: $-x^2 - 4x^3 - 9x^4 - \ldots$. Dodajmy je:

$$\begin{array}{cccccccccc} A & = & 0 & +1x & +4x^2 & +9x^3 & +16x^4 & +\ldots & \\ -xA & = & & & -1x^2 & -4x^3 & -9x^4 & -\ldots & \\ \hline A - xA & = & & +1x & +3x^2 & +5x^3 & +7x^4 & +\ldots & = B \end{array}$$

Wiemy, że $A - xA = A(1-x) = B$, a więc $A = \frac{B}{1-x}$. Powtórzmy poprzedni krok jeszcze raz, tym razem aplikując zmiany do B i uzyskując $-xB$:

$$\begin{array}{ccccccccc} B & = & 1x & +3x^2 & +5x^3 & +7x^4 & +\ldots & \\ -xB & = & & -1x^2 & -3x^3 & -5x^4 & -\ldots & \\ \hline B - xB & = & 1x & +2x^2 & +2x^3 & +2x^4 & +\ldots & = B \end{array}$$

Otrzymaliśmy więc funkcję $B(1-x) = x + 2x^2 + 2x^3 + 2x^4 + \ldots$. Znajdźmy dla takiego ciągu funkcję tworzącą, a następnie dla ciągu B:

$$B(1-x) = x + \frac{2x^2}{1-x} = \frac{2x^2 + x - x^2}{1-x} = \frac{x(1+x)}{1-x} \Longrightarrow B = \frac{x(1+x)}{(1-x)^2}$$

Wynik B możemy podstawić do wzoru na A, stąd więc mamy $A = \frac{B}{1-x} = \frac{x(1+x)}{(1-x)^3}$.

Przykład 2: W tym przykładzie chcemy uzyskać funkcję tworzącą ciągu $\langle n^3 \rangle = \langle 0, 1, 8, 27, \ldots \rangle$. Podobnie jak w pierwszym przykładzie, interesować nas będą pochodne (tam inaczej rozpisane). Tym razem potrzebna jest nam pochodna ciągu $\langle n^2 \rangle$, a następnie przesunięcie całego ciągu w prawo.

$$x \cdot A'(x) = x \cdot \frac{x^2 + 4x + 1}{(x-1)^4} = x \cdot (1 + 2^3 x^2 + 3^3 x^3 + 4^3 x^4 + \ldots) = \left\langle 0, 1^3, 2^3, 3^3, \ldots \right\rangle$$

Przykład 3: Teraz szukamy funkcję tworzącą ciągu $\left\langle \binom{n+k}{k} \right\rangle$. Kolejnymi wyrazami tego ciągu są $\left\langle 1, \binom{k+1}{k}, \binom{k+2}{k}, \binom{k+3}{k}, \ldots \right\rangle$. Chcemy obliczyć funkcję tworzącą

$$Z(x) = \sum_{k=0}^{\infty} \binom{n+k}{k} x^k$$

Wiemy, że szereg kolejnych dwumianów Newtona można zwinąć do zwięzłej funkcji:

$$\sum_{k=0}^{\infty} \binom{n}{k} x^k = (1+x)^n$$

Skorzystam również z poniższej własności symbolu Newtona:

$$\binom{n}{k} = (-1)^k \binom{-n+k-1}{k} \Longrightarrow \binom{n+k}{k} = (-1)^k \binom{-n-1}{k}$$

Łącząc powyższe informacje, przekształćmy wyjściową funkcję tworzącą $Z(x)$, aby otrzymać jej zwięzłą formę. Korzystamy najpierw z własności symbolu Newtona, później mnożymy $(-1)^k \cdot x^k = (-x)^k$. Na końcu zwijamy wszystko do jednej sumy, używając w tym celu uogólnionego symbola dwumianowego.

$$Z(x) = \sum_{k=0}^{\infty} \binom{n+k}{k} x^k = \sum_{k=0}^{\infty} (-1)^k \binom{-n-1}{k} x^k = \sum_{k=0}^{\infty} \binom{-(n+1)}{k} (-x)^k =$$
$$= (1 + (-x))^{-(n+1)} = \frac{1}{(1-x)^{n+1}}, \text{ co kończy dowód.}$$

Zadanie 6
Niech Q_k oznacza graf k-wymiarowej kostki, tzn. zbiór wierzchołków tego grafu tworzą wszystkie k-elementowe ciągi zer i jedynek i dwa wierzchołki są sąsiednie tylko wtedy, gdy odpowiadające im ciągi różnią się dokładnie jedną współrzędną. Oblicz, ile wierzchołków i krawędzi ma graf Q_k.

Obliczenie liczby wierzchołków grafu Q_k jest łatwe - są to wszystkie możliwe permutacje k-bitowych słów na słowniku $\Sigma = \{0, 1\}$, więc

$$|V| = 2^k$$

Aby obliczyć liczbę krawędzi grafu Q_k, musimy wziąć pod uwagę to, że ten graf jest nieskierowany, a więc (u, v) i (v, u) to jedna krawędź (dlatego dzielimy liczbę wszystkich krawędzi przez 2). Z każdego k-bitowego wierzchołka można wyprowadzić k krawędzi, np. dla 4-bitowego wierzchołka 0001 będą to 1001, 0101, 0011 i 0000. Mamy więc

$$|E| = \frac{k \cdot 2^k}{2} = k \cdot 2^{k-1}$$

Zadanie 7
Problem izomorfizmu dwóch grafów jest trudny. Załózmy natomiast, że w komputerze są dane dwa grafy G i H, określone na tym samym zbiorze wierzchołków $V(G) = V(H) = \{1, 2, 3, \ldots, n\}$. Niech m oznacza liczbę krawędzi grafu G. Podaj algorytm sprawdzający w czasie $O(m + n)$, czy te grafy są identyczne.

```
def is_identical(G, H):
    visited = [False, ..., False] # length n

    for vertex in vertices:
        nodes_diff = 0

        for neighbour in G[vertex]:
            visited[vertex] = True
            nodes_diff += 1

        for neighbour in H[vertex]:
            if visited[vertex] == False:
                return false

            nodes_diff -= 1

        if nodes_diff != 0:
            return false

    return true
```

Algorytm polega na tym, że dla każdego wierzchołka ze zbioru V dla grafu G sprawdzamy, czy istnieje taki sam sąsiad w grafie H. Ponadto w tablicy `visited` zapisujemy informacje o tym, czy wierzchołek był już odwiedzony. Zadaniem `nodes_diff` jest porównanie stopni wierzchołka `vertex` w każdym grafie - jeżeli liczba ta jest różna od 0, to grafy nie mogą być identyczne. W przeciwnym wypadku, po sprawdzeniu wszystkich wierzchołków, grafy na pewno są identyczne.

Zadanie 8

Rozważ reprezentacje grafu G: macierzową (za pomocą macierzy sąsiedztwa), listową. Dla każdej z tych reprezentacji określ złożoność wykonania na grafie G nastepujących operacji:

1. oblicz stopień ustalonego wierzchołka,
2. przeglądnij wszystkie krawędzie grafu,
3. sprawdź, czy krawędź (u, v) należy do grafu G,
4. usuń z grafu G krawędź (u, v),
5. wstaw do grafu G krawędź (u, v).

Operacja	Repr. macierzowa	Repr. listowa
obliczenie $\deg(v)$	$O(n)$	$O(\deg(v))$
przejrzenie krawędzi E	$O(n^2)$	$O(n+m)$
czy krawędź należy do G	$O(1)$	$O(\deg(u))$
usuń krawędź z grafu G	$O(1)$	$O(\deg(u))$
dodaj krawędź do grafu G	$O(1)$	$O(\deg(u) + \deg(v))$

1. Aby obliczyć stopień wierzchołka v należy przejrzeć cały wiersz w reprezentacji macierzowej i dodać elementy o wartości 1 (istniejący sąsiad), a więc zajmie to $O(n)$ czasu, a dla listy będzie to $O(\deg(v))$, gdyż w reprezentacji listowej przechowywani są sąsiedzi wierzchołka v, których jest tyle, ile wynosi $\deg(v)$.

2. Przejrzenie wszystkich krawędzi wymaga przejrzenia całej macierzy, więc $O(n^2)$, a w reprezentacji listowej listy sąsiadów każdego wierzchołka, czyli $O(n+m)$.

3. Przynależność krawędzi do grafu zajmuje czas $O(1)$ w reprezentacji macierzowej, wystarczy sprawdzić, czy np. `E[i][j] == 1`. W przypadku reprezentacji listowej należy sprawdzić całą listę sąsiadów, więc zajmuje to $O(\deg(u))$.

4. Usunięcie krawędzi z G zajmuje czas $O(1)$ w przypadku macierzowej reprezentacji, podobnie jak sprawdzenie przynależności do grafu, zmieniamy tylko wartość odpowiedniej komórki z 1 na 0. W przypadku reprezentacji listowej będzie to $O(\deg(u))$.

5. Dodanie krawędzi do grafu w reprezentacji macierzowej działa tak samo jak usuwanie i sprawdzenie przynależności do grafu G, więc zajmuje $O(1)$. W przypadku reprezentacji listowej jest to $O(\deg(u))$ dla grafu skierowanego lub $O(\deg(u)+\deg(v))$ dla grafu nieskierowanego (dodajemy krawędź do obu list sąsiadów).