Zadanie 1)

$A(x)$ - funkcja tworząca ciągu a_n. Podaj postać ~~dla~~ funkcji tworzącej dla ciągu $s_n = a_0 + a_1 + \ldots + a_n$

$\langle a_n \rangle = (a_0, a_1, a_2, \ldots, a_n, \ldots)$

$A(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \ldots = \sum_{i=0}^{\infty} a_i x^i$

z sumy ciągu geometrycznego: $\frac{1}{1-x} = 1 + x + x^2 + \ldots$

$A(x) \cdot \frac{1}{1-x} = a_0(1+x+x^2+\ldots) + a_1 x(1+x+x^2+\ldots) + a_2 x^2(1+x+x^2+\ldots) + \ldots$

$a_0 \quad a_0 x \quad a_1 x \quad a_0 x^2 \quad a_1 x^2 \quad a_2 x^2 \quad \ldots$

wyłączam składniki z $x^0, x^1, x^2, \ldots$

$= \underset{s_0}{a_0} + x\underset{s_1}{(a_0 + a_1)} + x^2\underset{s_2}{(a_0 + a_1 + a_2)} + \ldots$

$= \sum_{i=0}^{\infty} a_i s_i$ Zatem $A(x)\frac{1}{1-x}$ jest funkcją tworzącą dla ciągu s_n

Zadanie 6)

Liczba wierzchołków to 2^k (ponieważ jest 2^k ciągów długości k składających się z 0 i 1)

Każdy wierzchołek ma k sąsiadów (ponieważ jeśli mamy ciąg długości k, to mamy k możliwości zmiany pojedynczego bitu)

Nasz graf jest nieskierowany, zatem łączna liczba krawędzi wynosi

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

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

Zadanie 7)

Algorytm dla każdego wierzchołka sprawdza jego listę sąsiedztwa w grafie G i H. W tablicy visited[] sprawdzamy, czy w obu grafach spojrzymy na te same wierzchołki. Ponadto stopnie wierzchołków muszą się zgadzać. Zwracamy fałsz, jeśli w którymkolwiek momencie znajdziemy sprzeczność. Jeśli dojdziemy do końca to zwracamy prawdę

```
def sprawdź(G, H):
    visited = [0, 0, ..., 0]   # tablica z zerami o długości |V|
    for v in vertices:
        d=0
        for u in G[v]:
            visited[u] = 1     # każdemu sąsiadowi (v) zaznaczam 1 i
            d += 1             # zwiększam d (stopień)
        for u in H[v]:
            if visited[u] == 0:   # jeśli w H wierzchołek (v) ma innego
                return False      # sąsiada to sprzeczność
            visited[u] = 0     # odznaczam visited
            d -= 1
        if d != 0:             # złe stopnie, sprzeczność
            return False

    return True                # OK
```

8)

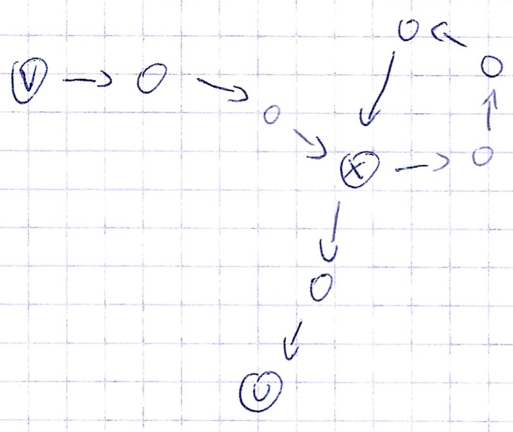
	macierz	lista
$\deg(v)$	$O(n)$	$O(1)$ jeśli możemy sprawdzić rozmiar listy $O(\deg(v))$ jeśli nie możemy
przejrzenie wszystkich krawędzi	$O(n^2)$	$O(n+m)$
sprawdzenie czy $(v,u) \in G$	$O(1)$	$O(\deg(v))$
usuń krawędź	$O(1)$	$O(\deg(v))$
dodaj krawędź	$O(1)$	$O(1)$

Zadanie 9)

Mamy drogę $v \rightsquigarrow u$ (krawędzie co najwyżej raz, wierzchołki mogą być kilka razy)

Chcemy ścieżkę $v \rightsquigarrow u$ (krawędzie i wierzchołki co najwyżej raz)

Jeśli droga nie jest ~~ścież~~ ścieżką, to istnieje taki wierzchołek x, który występuje tam więcej niż raz. Musi istnieć tam cykl, który możemy usunąć. Powtarzając tę operację, dopóki istnieje wierzchołek występujący więcej niż raz, otrzymamy ścieżkę poprzez usunięcie wszystkich cykli



$\Longrightarrow$