5.

5. Zmodyfikuj algorytm Dijkstry tak, by działał dla grafów skierowanych.

$G = (V, E)$ - graf spójny; $c : E \to R \geq 0$, $s \in V$

$d(v)$ - waga najkrótszej ścieżki z s do v
$t(v)$ - waga najkrótszej prawie S-owej ścieżki z s do v; jeśli takiej scieżki nie ma, to $t(v) = \infty$

$S \leftarrow \{s\}$, $d(s) \leftarrow 0$
dla każdego sąsiada v wierzchołka s: $t(v) \leftarrow c(s, v)$
dla pozostałych wierzchołków: $t(v) \leftarrow \infty$
dopóki $S \neq V$ wykonaj:
    $u \leftarrow \; argmin\{t(u) : u \notin S\}$
    dodaj u do S
    zaktualizuj wartości $t(v)$:
    dla każdego sąsiada $v \notin S$ wierzchołka u: (1)
    $t(v) \leftarrow \min\{t(v), d(u) + c(u, v)\}$

Algorytm Dijkstry działa dla grafów skierowanych. Sprawdzając sąsiadów (1) sprawdza on wierzchołki do których może się dostać z u. Działa to identycznie dla grafów nieskierowanych i skierowanych

6. Droga Eulera:

- Warunek wystarczający i konieczny:

  Co najwyżej jeden wierzchołek ma stopień wejściowy o 1 mniejszy niż wyjściowy oraz co najwyżej jeden wierzchołek ma stopień wejściowy o 1 większy niż wyjściowy.
  Każdy inny wierzchołek ma te stopnie równe.

Cykl Eulera:

- Warunek wystarczający i konieczny:

  każdy wierzchołek ma stopień wejściowy równy wyjściowemu

Algorytm znajdujący cykl Eulera: (Idea: Algorytm Hierholzera)

1. Sprawdzamy warunek wyżej, jeżeli nie zachodzi to nie ma cyklu Eulera

2. Wybieramy dowolny wierzchołek v

3. Stos $\leftarrow \{v\}$, cykl $= \{\}$

4. Dopóki Stos $\neq \emptyset$:

   5. $u \leftarrow$ stos.top() // zwraca wierzchołek z góry stosu

   6. Jeżeli nie istnieje żadna krawędź wychodząca z u:

      - stos.pop() // usuwamy u
      - cykl += {u}

      Wpp: Weź dowolną krawędź wychodzącą z u.
      Wierzchołek x do którego ta krawędź wchodzi
      wrzuć na stos. Usuń krawędź (u, x) z G

7. return cykl.reverse

7. Weźmy najdłuższą ścieżkę P w G oraz ostatni wierzchołek tej ścieżki, nazwijmy go v. Wszyscy sąsiedzi v muszą znajdować się na ścieżce P.

Inaczej P nie byłaby najdłuższa.

3 sąsiedzi v: x, y, z

$v-y$, $v-x-y$, $v-z-y$

Ścieżki te mają taki sam początek i koniec więc połączenie 2 z nich da cykl, np: $v-y-x-v$.

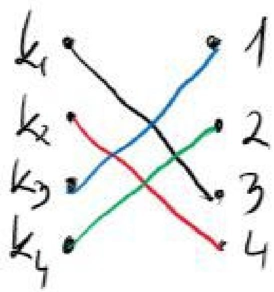
Z zasady szufladkowej Dirichleta, dwie z tych ścieżek mają tą samą parzystość. Suma dwóch liczb o tej samej parzystości daje liczbę parzystą.

10. Rozważmy graf dwudzielny w którym jeden podzbiór V to indeks kolumny (k_i: i-ta kolumna) a drugi to liczby od 1 do n.
Oba podzbiory mają rozmiar n.

Wierzchołki są połączone wtedy gdy danej liczby x nie ma w kolumnie k_i

Przykład:

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 4 & 1 & 2 & 3 \\ 2 & 3 & 4 & 1 \end{bmatrix} \rightarrow$$



Aby rozszerzyć prostokąt Łaciński musimy znaleźć skojarzenie doskonałe takiego grafu.

Aby rozszerzyć prostokąt łaciński musimy znaleźć skojarzenie doskonałe takiego grafu.

Wierzchołek k_i jest połączony z liczbami których w tej kolumnie nie ma. Takich liczb jest $n-m$. Zauważmy również że liczba x występuje wyłącznie m razy, a więc istnieje $n-m$ kolum w których nie ma liczby x. To gwarantuje nam że takie skojarzenie doskonałe zawsze istnieje.