

Звіт з лабораторної роботи №2  
з дисципліни «Комп'ютерна графіка»  
на тему «Локалізація точки на планарному розбитті: методом ланцюгів»  
студентки 3-го курсу Факультету комп'ютерних наук та кібернетики  
групи ІПС-32  
Бондарець Дарини Володимирівни

**Постановка задачі.**

Задано плоский прямолінійний граф (ППЛГ), та деяка точка  $Z$ . Локалізувати точку (вказати характеристику місцезнаходження на площині) на цьому розбитті.

Розбити граф на ланцюги, та вказати між якими двома найближчими ланцюгами лежить вказана точка.

Вершини графа задані своїми координатами, ребра задані матрицею суміжності графа.

**Розв'язання.**

Виконання алгоритму має дати відповідь на питання між якими двома ланцюгами графа лежить точка, у разі якщо точка належить графу, і в іншому разі – повідомити, що точка лежить поза графом.

Нехай задано регулярний ППЛГ. Надалі необхідно виконати три основні процедури: балансування ребер за вагою, розбиття ребер на ланцюги, пошук серед ланцюгів.

Отже, занумеруємо вершини у порядку зростання її ординати, якщо ординати вершин рівні, впорядкуємо за зростанням абсциси. Для кожної вершини визначимо впорядковані множини ребер, що входять та виходять з них, причому вхідні ребра ( $IN(v_i)$ ) впорядковані за кутом проти годинникової стрілки, а вихідні ребра ( $OUT(v_i)$ ) – за годинниковою стрілкою.

Позначимо як  $W_{IN}(v)$  – сумму вагів вхідних ребер, за  $W_{OUT}(v)$  – сумму ваг вихідних. Необхідно зважити та збалансувати ребра графа.

Алгоритм балансування **balance(G):**

1. for each edge “e” do  $W(e) = 1$  (initialization)
2. for  $i:=2$  until  $N-1$  do{  
    підрахувати  $W_{IN}(v)$ ;  
     $d$  – крайнє зліва ребро, яке виходить з  $v_i$ ;  
    if ( $W_{IN}(v) > OUT(v_i)$ ) then  $W(d) = W_{IN}(v_i) - OUT(v_i) + 1$   
  }  
  for  $i:=N-1$  until 2 do{  
    підрахувати  $W_{OUT}(v)$ ;  
     $d$  – крайнє зліва ребро, яке входить в  $v_i$ ;  
    if ( $W_{OUT}(v) > IN(v_i)$ ) then  $W(d) = W_{OUT}(v_i) - W_{IN}(v_i) + W(d)$   
  }  
}

Розбиваємо граф на ланцюги на один прохід по усім ребрам таким чином:

створюємо стільки пустих списків, які представляють ланцюг, скільки вихідних ребер має найменша ( $\min\{(x,y)\}$ ) вершина.

Проходимо по впорядкованому від початку роботи списку вершин, починаючи з найменшої стільки разів, скільки витоків (вихідних ребер) вона має. Обираємо кожного разу найлівіше (найменше не використане ребро у впорядкованій множині ребер вершини) ребро, що виходить з даної вершини та додаємо його до поточного ланцюга, продовжуємо процедуру з наступними вершинами таким же чином, поки не доходимо до найбільшої вершини. Кожного разу від ваги доданого до деякого ланцюга ребра віднімається 1, якщо вага ребра 0, воно вважається використаним, тому більше не може додаватися до жодного ланцюга.

Після виконання процедури розбиття отримуємо впорядкований список ланцюгів, у якому можна здійснюємо двійковий пошук за критерієм «зліва чи справа від ланцюга знаходиться точка» та локалізуємо точку.

Відповідні складності передобробки та виконання можуть складати:

$O(N * \log(N))$  – для регуляризації графа.

$O(N)$  – для розбиття на ланцюги.

$O(\log(p) * \log(r))$  – для локалізації,  $p$  – к-ть вершин у ланцюзі,  $r$  – к-ть ланцюгів.

**Мова реалізації:** Java