УДК 316.77

МЕТОДИ АВТОМАТИЧНОЇ ГЕНЕРАЦІЇ ГРАФІВ ЗІ ЗВ'ЯЗАНИМИ ВЕРШИНАМИ

Хліста Руслан

Кіровоградський машинобудівний коледж Кіровоградського національного технічного університету

(Кропивницький)

Анотація. У статті запропоновані алгоритми автоматичної генерації: зв'язного неорієнтованого дерева та простого зв'язного неорієнтованого графа. Для алгоритму генерації зв'язного неорієнтованого дерева на вхід обчислювального пристрою подається кількість вершин, на виході буде отримана матриця суміжності дерева у вигляді двовимірного масиву цілих чисел. Для алгоритму генерації графів на вхід подається матриця суміжності дерева, на виході буде отримано матриця суміжності неорієнтованого графа. Алгоритми призначені для систем автоматичної генерації задач та тестових завдань з дискретної математики, математичної логіки та інформатики. Також розроблені алгоритми можуть бути застосовні для дослідження переміщення мобільного агенту по вершинах графа, моделювання роботи інформаційних мереж та інших завдань, які використовують апарат графів.

Ключові слова: алгоритм, дерево, граф, генерація, матриця суміжності.

Постановка проблеми. Незважаючи на те, що будувати випадкові графи потрібно при вирішенні багатьох прикладних задач, більшість авторів розглядають рішення задачі генерації цих структур як очевидне. Проте, при практичній реалізації алгоритмів виникає безліч «підводних каменів», пов'язаних з часом виконання алгоритмів на обчислювальному пристрої, простотою реалізації програмного коду при використанні алгоритму тощо [1], [2], [4]. Тому проблема розробки алгоритмів генерації графів, що виконують свою роботу швидко і ефективно, є актуальною і потребує розгляду.

Аналіз актуальних досліджень. Проблемою автоматичної генерації графів переймались багато дослідників. Це питання так чи інакше підіймалось в працях Р. Хаггарти [9], Т.Х. Кормена [7], Г. Дудека [1], С. Родіонова[5], М. Дженкинса[1], Б. Купера [2], Х. Чо[6], С. В. Сапунова [3] та ін. Цими авторами була розроблена велика кількість алгоритмів для генерування графів різних типів. Але запропоновані алгоритми мали або складну структуру, або невизначений час роботи при реалізації їх на обчислювальному пристрої.

Метою статті є розробка алгоритмів генерування зв'язних графів для автоматичного створення завдань з дискретної математики і інформатики, які можна було б використовувати в навчальному процесі при розробці задач, тестів, контрольних робіт тощо.

Методи досліджень полягають в теоретичному дослідженні літератури з проблеми генерування графів різних типів та аналізу запропонованих там алгоритмів.

Виклад основного матеріалу. Всі поняття, що не визначаються загальновідомі і можуть бути знайдені, наприклад, в [7].

Простий зв'язний кінцевий неорієнтований граф будемо позначати $G(V,E)$, де V – кінцева множина вершин, E – множина ребер. Надалі, якщо не зазначено інше, в роботі розглядаються саме такі графи. Деревом називається граф без циклів. Матрицею суміжності графа G з n вершинами називається квадратна матриця A порядку n, у якій елемент A_{ij}, який знаходиться на перетині рядка з номером i і стовпця з номером j, визначається наступним чином: $A_{ij}=1$, якщо $(i,j)\in E$ і $A_{ij}=0$, якщо $(i,j)\notin E$.

Відстанню між двома вершинами графа називається довжина найкоротшого шляху, що з'єднує ці вершини. Відстань від даної вершини v до найбільш віддаленої від неї вершини називається ексцентриситетом вершини v. Величина найбільшого ексцентриситету називається діаметром графа.

Базовий алгоритм генерації дерева

В запропонованому методі генерації графа першою задачею є побудова випадкового дерева з заданою кількістю вершин. Тут $G(V,E)$ позначає неорієнтоване зв'язне дерево. Далі будем вважати, що B – множина незв'язних вершин; C – множина зв'язних вершин; $A[G]$ – матриця суміжності $G(V,E)$ виду $A[i,j]$, де $i\in[0\ldots n]$, $j\in[0\ldots n]$, $n\in N$ – кількість вершин $G(V,E)$.

Алгоритм 1. Базовий алгоритм генерації дерева

Вхід. n

Вихід. $A[G]$

Метод.

1. Ініціалізувати нулями матрицю $A[G]$, $C:=\varnothing$, створити множину B з n натуральних чисел (незв'язних вершин);
2. Обрати і видалити з B довільну вершину s_0 ($B:=B\setminus\{s_0\}$);
3. Помістити s_0 у C ($C:=C\cup\{s_0\}$) ;
4. Якщо $B=\varnothing$, то шукане дерево G вже побудовано, і алгоритм закінчує роботу, інакше перейти до п.5;
5. Обрати довільну вершину $s_i\in B$;
6. Обрати довільну вершину $s_j\in C$;
7. $A[i,j]:=1$, $A[j,i]:=1$;

8. $B := B \setminus \{s_0\}$, $C := C \cup \{s_0\}$;

9. Перейти до п. 4.

Теорема 1. *Алгоритм 1 будує дерево G з n-вершинами за N кроків*

Доведення

Так як множина B кінцева, і на кожній ітерації алгоритму його потужність зменшується на 1, то алгоритм завершує роботу за n кроків.

На кожній ітерації алгоритму вершина, що обробляється, з'єднується ребром з вершиною, що належить до зв'язного графу, отже, результуючий граф є зв'язним.

Кожного разу при додаванні нового ребра його кінцями є вершина дерева з множини C і вершина, що цьому дереву не належить (з множини B). Таким чином, в результаті додавання ребра не виникає цикл. З цього слідує, що результуючий граф є деревом.

Що і треба було довести.

Виконаємо деталізацію алгоритму 1 у вигляді блок-схеми. При проектуванні блок-схеми будемо використовувати наступні позначення: $b[]$ – масив незв'язних вершин; $c[]$ – масив зв'язних вершин; $Rand(t,p)$ – функція, що генерує випадкове ціле число з проміжку $[t, t+1, \ldots p]$, $t \le p$, генератор дає рівномірний розподіл вірогідності; $INC()$, $DEC()$ – оператор інкременту ті декременту; k – випадково обраний номер елементів масиву незв'язних вершин; q – випадково обраний номер елементів масиву зв'язних вершин; m – кількість незв'язних вершин.

Блок-схема базового алгоритму генерації дерева представлена на рис. 1.

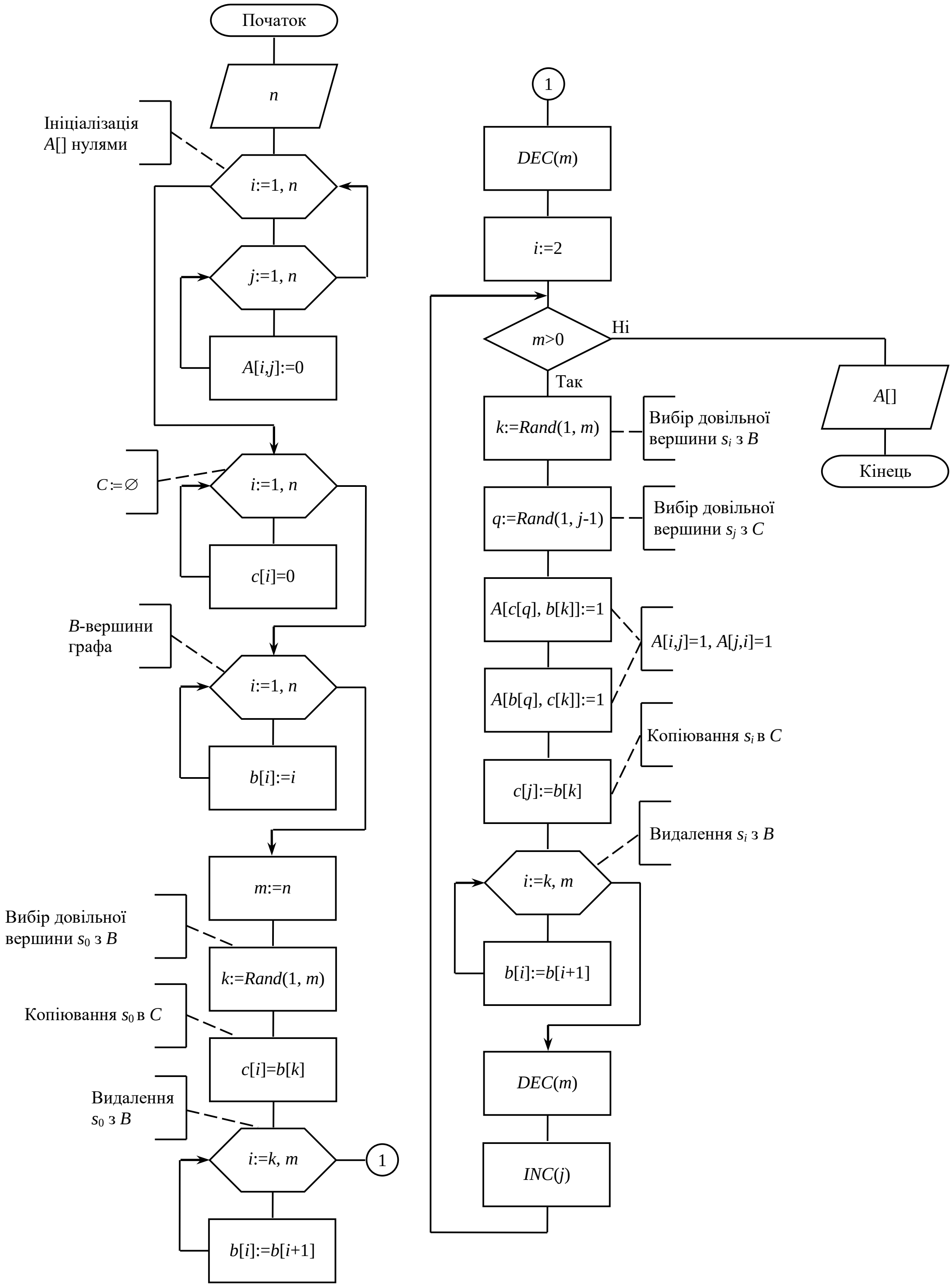

Рис. 1. Блок-схема базового алгоритму генерації дерева

Базовий алгоритм генерації зв'язного графа

Так як в роботі розглядаються зв'язні неорієнтовані графи, то для обчислень можна використовувати тільки верхній трикутник матриці суміжності. Тут $G'(V,E)$ позначає неорієнтований зв'язний граф, де V – кінцева множина вершин, а E – множина ребер; $nc \in N$ – кількість ребер для формування циклів; FV – множина, елементи якої є пари координат з верхнього трикутника $A[G]$ (під координатами розуміється номер рядка і стовпця матриці); $length[D]$ – функція, яка підраховує кількість елементів множини D.

Алгоритм 2. Базовий алгоритм генерації зв'язного графа

Вхід. $A[G]$, nc

Вихід. $A[G']$

Метод.

1. Якщо $nc > (n^2 - 3n + 2)/2$, то шуканий граф згенерувати неможливо і алгоритм закінчує роботу, інакше перейти до п. 2;
2. Ініціалізувати FV координатами відсутніх ребер (нулів) з верхнього трикутника $A[G]$;
3. $E_p := length[FV]$;
4. Якщо $nc > 0$, то перейти до п. 5, інакше шуканий граф вже сформовано, і алгоритм закінчує роботу;
5. $E_{ran} := Rand(1, E_p)$;
6. $A[FV[E_{ran}]] := 1$;
7. Видалити $FV[E_{ran}]$ з FV;
8. $DEC(E_p)$, $DEC(nc)$;
9. Перейти до п. 4;

Теорема 2. *Алгоритм 2 будує зв'язний граф G' з n вершинами за час* $O(n^2)$.

Доведення

Так як на кожній ітерації алгоритму значення nc зменшується на 1, то алгоритм зупиняється при $nc = 0$.

На вхід алгоритму подається дерево G'. Під час роботи алгоритму додаються нові ребра и не додаються нові вершини. З цього слідує, граф G' є зв'язним.

Так як nc не перевищує $\left(n^2 - 3n + 2\right)/2$, то число ітерацій циклу дорівнює $O(n^2)$.

Що і треба було довести.

При проектуванні блок-схеми використані наступні позначення: $FV[]$ – масив, даними якого є елементи структури (Row, Col), де $Row, Col \in N$ і однозначно визначають ребра графа; $Del(D, i)$ – процедура, яка видаляє елемент $i \in D$ з множини D.

Блок-схема базового алгоритму генерації графа представлена на рис. 2.

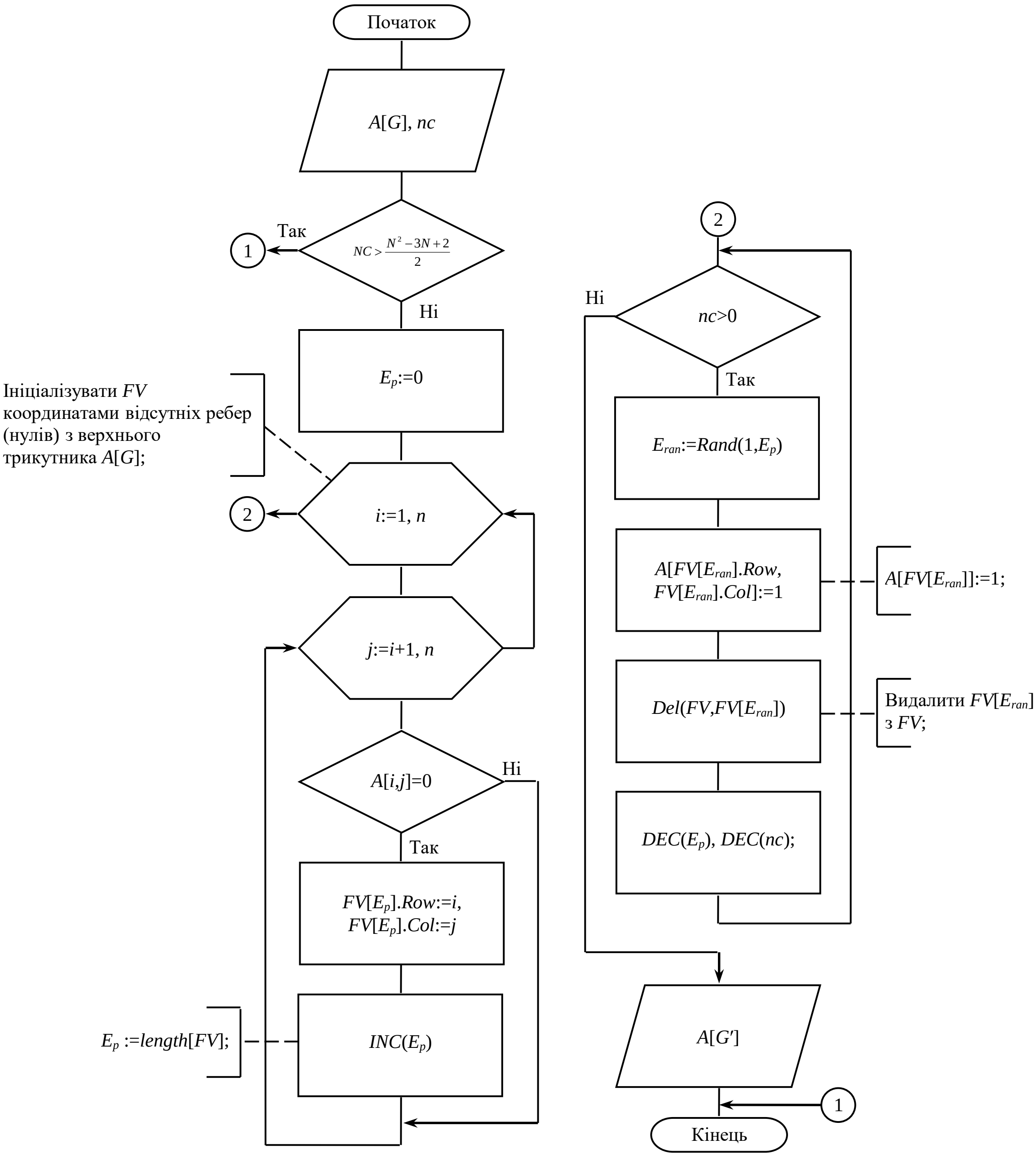

Рис. 2. Блок-схема базового алгоритму генерації графа

Висновки. В роботі побудовані алгоритми генерації матриці суміжності неорієнтованого дерева з фіксованим числом кроків в ітерації, алгоритм генерації матриці суміжності зв’язного графа Також наведені блок-схеми цих алгоритмів, які пояснюють їх роботу.

Отримані алгоритми можуть бути використані при автоматичному генерування графів для розробки завдань з дискретної математики, інформатики та як поля для дослідження переміщення мобільного агенту по вершинах графу [3], [8], моделювання роботи інформаційних мереж [10] та інших задач, що використовують апарат графів.

Перспективи подальших наукових розвідок пов'язані з пошуком оптимальних алгоритмів автоматичної генерації графів з заданими параметрами.

БІБЛІОГРАФІЯ

1. Dudek G., Jenkin M. Computational Principles of Mobile Robotics – Cambridge: Cambridge University Press, 2010. – 391 p.

2. Kuipers B. The Spatial Semantic Hierarchy // Artificial Intelligence. – 2000. – Vol. 119. – P. 191-233.

3. Rodionov S., and Choo H. "On Generating Random Network Structures: Trees," Lecture Notes in Computer Science, Vol. 2658, pp. 879-887, 2003.

4. Rodionov S., and Choo H. "On Generating Random Network Structures: Connected Graphs," Lecture Notes in Computer Science, Vol. 2658, pp. 483-491, 2003.

5. Waxman B.M., "Routing of Multipoint Connections," IEEE JSAC, Vol. 9, pp. 1617-1622, 1993.

6. Грунский И.С. Диагностика местоположения мобильного робота на основе топологической информации о среде / И. С. Грунский, С. В. Сапунов – Искусственный интеллект. – 2011. – №2. – С. 15-25.

7. Касьянов В.Н. Графы в программировании: обработка, визуализация и применение / Касьянов В.Н., Евстигнеев В.А. – СПб.: БХВ-Петербург, 2003. – 1104 с.

8. Кормен Т.Х. Алгоритмы: построение и анализ / Кормен Т.Х. – М.: Издательский дом «Вильямс», 2005. – 2-е изд. – 1296 с.

9. Сапунов С.В. Определение положения робота в топологической среде / Сапунов С.В. – Искусственный интеллект. – 2008. – №4. – С. 558-565.

10. Хаггарти Р. Дискретная математика для программистов / Р. Хаггарти – Москва: Техносфера, 2003. – 320с.

Відомості про автора

Хліста Руслан Олександрович – викладач І категорії Кіровоградського машинобудівного коледжу Кіровоградського національного технічного університету.

Наукові інтереси: методика навчання математики, програмування у вищих навчальних закладах І-ІІ рівня акредитації.