

ОНАГЛЕДЯВАНЕ НА АЛГОРИТМИТЕ ЗА ОБХОЖДАНЕ НА ГРАФ

Богдан Методиев Христов

Представени са идеи, реализирани в програмен продукт, който: въвежда граф от текстов файл или чрез изобразяване на върхове и ребра от потребителя върху екрана. Същият демонстрира чрез анимация двете стратегии за обхождане на граф и някои техни приложения: при обхождане в дълбочина (съществуване на път, намиране на един или всички пътища, проверка за наличие на цикъл) и при обхождане в ширина (определяне на минимален път, алгоритъм на Дейкстра, създаване на минимално покриващо дърво). Програмният продукт е предназначен за онагледяване на съответните теми в обучението по информатика (профилирана подготовка) в средното училище, но може да се използва с успех и във висшите училища. Разработен е в средата Microsoft Visual Studio.NET 2003 с използване на програмния език C#.

1. Основание. Съгласно действащата учебна програма по информатика – XI–XII клас, второ равнище, учениците трябва да могат да решават задачи в граф, чрез обхождане в ширина или дълбочина. За проявяващите интерес и желание да участват в олимпиади по информатика, се препоръчват следните допълнителни теми: построяване на минимално покриващо дърво, намиране на най-къс път в граф, решаване на задача в граф по схемата backtracking. Изброените теми изискват сериозно мислене и задълбочаване, но често остават неразбрани от учениците. Необходим са допълнителни усилия за разкриване на тяхната същност. В почти всички литературни източници се задоволяват с описание на предизвикваните действия от изпълнението на съответния код, т.е. първо се вижда кода на готово, а след това се прави опит за разясняване. В [1] преобладава подходът, при който се тръгва от практически пример, конкретизира се поставената задача, анализират се различните възможни случаи и се достига до формулиране на обобщен алгоритъм. Разсъжденията на автора (Т. Смит) се придружават от схеми, чертежи, словесни описания на отделни стъпки от алгоритъма. Нашето желание е да се направи и следващата крачка: **да се демонстрира нагледно движението в графа чрез използване на съответен програмен продукт** (не ни е известно да има създадена такава програма). По този начин учениците ще могат зрительно (и слухово) да възприемат и осъзнават преподаваното. Те ще наблюдават постъпковото придвижване в графа, а учителят ще спира тяхното внимание на съществените моменти. Демонстрациите могат да се повторят няколкократно и да се обхванат достатъчен брой частни случаи, които понякога се пропускат и от добри специалисти.

2. Предварителна подготовка. Необходимо е учителят да подготви няколко конкретни примера за съответния урок. В литературата трудно се намират практически задачи – опитът показва, че именно те мобилизират мисленето на учениците. В [2] обикновено входните данни се задават като константи (матрица на съседство) в предложената програмна реализация. Ние основно използваме примера за маршрутите на авиокомпания от [1, стр. 387], но се отказваме от представянето на графа чрез свързани списъци и работим с масиви. Справочникът на върховете и матрицата на теглата трябва да се въведат предварително. Входните данни ги поставяме в текстов файл, който може да се създаде с обикновен текстов редактор. Използваме следната структура на файла: първият ред съдържа броя на върхове – n , в следващите n реда са данните за върховете, а в последните n реда е разположена матрицата на теглата/съседство. С цел бързо изобразяване на графа върху екрана е необходимо за всеки връх да се посочат компютърните му координати и да бъдат подредени във възходящ ред, като номерирането започва от нула (по желание може да се добави и име, свързано с върха). Алгоритъмът за въвеждане елементите на граф от текстов файл не е сложен и винаги го разглеждаме с учениците.

3. Основни моменти от интерфейса. Използва се меню със следните възможности: Работа с файлове, Редактиране, Изобразяване, Обхождане в дълбочина, Обхождане в ширина. Чрез Зареждане от файл се активира стандартна диалогова кутия и се избира желан текстов файл, съдържащ предварително въведени данни за граф. За стратегията обхождане в дълбочина са подготвени за онагледяване следните алгоритми:

- обхождане в дълбочина [2, стр. 259];
- съществуване на път;
- намиране на всички прости пътища [2, стр.265];
- проверка за ацикличност на граф [2, стр. 263];

При използване на стратегията обхождане в ширина се онагледяват отново четирите алгоритма:

- обхождане по нива [2, стр. 257];
- определяне път с минимален брой ребра [2, стр.262];
- алгоритъм на Дейкстра [2, стр. 273];
- създаване на минимално покриващо дърво с алгоритъма на Прим [2, стр. 318].

Избраните алгоритми са съобразени с учебната програма за профилирана подготовка. Като използвана литература преобладава [2], поради факта, че там графът се представя чрез матрица на теглата/съседство. В пряката си работа обаче използваме примери и методика на преподаване, изложени в [1]. Основният пример е този от стр. 387 и е представен на чертежа.

Предвидена е възможност да се въвежда граф чрез използване на мишка. За целта е предоставена лента с инструменти, чрез които: може да се добавя връх (след щракване с ляв бутон), да се изобразява еднопосочно/двупосочно/неориентирано ребро чрез изтегляне с мишката.

След като веднъж е изобразен графът, могат да се редактират неговите елементи: компютърните координати на мястото на изобразяване на върховете, теглата на ребрата, мястото на появяване на номерата или имената на върховете, може да се добави или премахне връх (добавяне или премахване на ребро се извършва чрез



въвеждане на ненулево или нулево тегло). След редактиране или интерактивно създаване на нов, графът може да се съхрани под същото или ново име на текстов файл.

С помощта на опцията Изобразяване може да се определи начина на изобразяване на графа: чрез върхове и/или ребра, с номера и/или имена на върхове, с тегла и/или посока на ребрата. За целта се използват полета за маркиране, като се допуска посочване на няколко възможности едновременно.

4. Анимиране на известните алгоритми. При обхождане в дълбочина визуализирането върху екрана е в следната последователност: всички върхове от графа се появяват в син цвят, текущия връх се изобразява с черен цвят, а кандидатите за следващ текущ – в червен. След това за следващ връх се избира кандидатът с най-малък номер, а на останалите се премахва червената маркировка. Следва изобразяване на реброто, свързващо стария текущ връх с новия такъв. При невъзможност да се продължи напред, се реализира връщане назад, като цвета на реброто постепенно става зелен. Между всяка една от изброените стъпки се прави пауза, чиято продължителност се определя от потребителя.

Интересено е визуализирането на обратния ход при рекурсията: без да е написан нито един ред програмен код, се осъществява постепенно оцветяване в зелен цвят на целия изминат път (до началния връх) в обратна посока. Този момент трудно се осмисля от учениците и онагледяването е крайно необходимо. При търсене и определяне на един път нещата се повтарят, с тази разлика, че процесът на обхождане се прекратява при достигане на върха, посочен като краен.

Когато се илюстрира намирането на всички пътища (тъй като вариантите са твърде много), при връщане назад се използва пълно изтриване на открития път отзад напред, до върха, който позволява намиране на ново продължение до крайния връх.

При проверката за ацикличност на граф, за всеки един от върховете се изобразява намерения цикъл, ако съществува такъв. Откриването на циклите става чрез

обхождане в дълбочина.

При обхождането в ширина се следва горната стратегия на визуализиране. Тъй като обхождането е по нива, целесъобразно е елементите от едно ниво (върхове и ребра) да бъдат оцветени с един цвят, различен от този в съседните нива. Освен това, при извеждане на резултата от обхождането, отново се използва подреждането на върховете по нива.

Пътят с минимален брой ребра се онагледява както обхождането в ширина, само че в крайния резултат пътят се извежда в обратен ред на посещаване на върховете, с цел по-лесно проследяване на пътя в графа.

При онагледяването на алгоритъма на Дейкстра отново се използва различно оцветяване по нива. При този алгоритъм съществено се използват теглата на ребрата и затова до всеки връх се изобразява минималното тегло на пътя от него до върха, посочен като начален. В края на алгоритъма се виждат пресметнатите тегла на пътищата от всеки връх до началния.

При определянето на минимално покриващо дърво чрез алгоритъма на Прим, поетапно се изобразяват теглата на ребрата, които отговарят на критерия за минималност. Намереното минимално покриващо дърво се изобразява в долния десен прозорец като списък от ребра.

5. Методически похвати. За всеки един от посочените по-горе алгоритми се отделят два учебни часа. В края на предходното занимание пред учениците се поставя като домашно задание, да се опитат да открият някакво правило за движение в графа, с което да се реши задача, която ще бъде разглеждана на следващото събиране. След обсъждане на предложените от учениците идеи, с помощта на програмния продукт се онагледява обсъждания алгоритъм. Могат да се направят няколко повторения с различни стойности на параметрите. Тези разсъждения е добре да се извършват при работа на един компютър и визуализиране чрез проектор. При обхождане в ширина задължително изобразяваме (върху черната дъска) графа като дърво с корен върха, посочен за начален. Следва колективно обсъждане на словесното описание на разглеждания алгоритъм и опит за описване на псевдокод. Уточняват се необходимите допълнителни променливи и масиви, техния вид и тип. До края на първия час се оставят учениците сами да опитат да напишат необходимия код. В началото на втория час колективно се записва кодът на алгоритъма върху черната дъска и в тетрадките на учениците. След това се въвежда във всеки един компютър. Учениците разполагат със собствени акаунти и защитено дисково пространство. При изпитване те винаги могат да ползват собствените си тетрадки и всичко, което са въвели до момента в компютрите си (включително и от предходните учебни години). Това е един съществен стимул за работа на учениците в редовните часове – може да не участват с идеи при колективните обсъждания, но в техни интерес е да въведат кода на алгоритъма и да го изчистят от синтактични грешки. При това, учениците винаги са съветвани при възможност да копират и да модифицират алгоритми, когато това е възможно. В края на изучаването на темите свързани с граф, учениците разполагат с две програми – менюта (конзолни приложения), чрез които се зарежда разглеждания граф от текстов файл и се стартира един от споменатите по-горе алгоритми.

ЛИТЕРАТУРА

- [1] Т. М. Смит. Принципи и методи на програмирането с Pascal. София, Техника, 2001.
- [2] П. Наков, П. Добриков. Програмиране = ++Алгоритми. София, Тор Team Co., 2003.

Богдан Методиев Христов
ул. Неофит Бозвели № 21
2500 Кюстендил
e-mail: hristov_b@abv.bg

EXEMPLIFYING GRAPH TRAVERSAL ALGORITHMS

Bogdan Metodiev Hristov

Some ideas are shown through a software product that: inserts a graph from a text file or through vertices and edges inserted by a user on the screen; using animation, shows the two strategies of graph traversal and some of their applications: depth-first traversal (existence of a path, finding one or all paths, controlling the existence of a cycle), and breadth-first traversal (finding the shortest path, Dijkstra's algorithm, creating a minimal covering tree). The software product is intended for exemplifying the corresponding themes in the specialized IT training in high schools, and can be used at the universities as well. The software is designed in Microsoft Visual Studio.NET 2003 environment with the language C#.