Лекція 14. Пошук оптимальних маршрутів у графі

План

1. ***Алгоритми пошуку маршрутів у графах. Теорема Таррі***
2. ***Алгоритми пошуку мінімальних маршрутів у ненавантаженому та навантаженому графах***

1. *Алгоритми пошуку маршрутів у графах. Теорема Таррі*

У багатьох практичних задачах, математичні моделі яких описують графами, потрібно відшукати маршрут, який з'єднує ці вершини. Для зв'язного графа такий маршрут завжди існує.

Алгоритм Таррі

Початкова вершина – V_p, кінець маршруту – V_k.

Для відшукання одного із маршрутів, який сполучає вершини V_p і V_k користуються правилом:

1. Пошук маршруту починають з вершини V_p, ідуть по довільному ребру, інцидентному цій вершині, відмічаючи напрямок у якому це ребро пройдене. Виходячи з вершини V_i вибирають те ребро, яке не було пройдене ще жодного разу. Якщо таких ребер не має, то вибирають ребро, яке було пройдене у протилежному напрямку.
2. Для кожної вершини відмічають те ребро (засікою) по якому у цю вершину зайшли перший раз. По ребру із засікою вертаються лише тоді, коли інших можливостей немає.

Через скінченне число кроків буде знайдена вершина V_k. Алгоритм Таррі результативний для довільного зв'язного графа. Якщо граф незв'язний, то алгоритм дозволяє виділити компоненти зв'язності.

Теорема Таррі. Якщо граф зв'язний, то можна побудувати циклічний маршрут, який містить усі ребра графа рівно два рази, по одному в кожному напрямку.

2. Алгоритми пошуку мінімальних маршрутів у ненавантаженому та навантаженому графах

Нехай задано неорієнтований зв'язний граф $G = (U, V)$ і дві його вершини V_p і V_k.

Означення. Маршрут з вершини V_p у вершину V_k називається **мінімальним**, якщо він з'єднує ці вершини і має найменшу довжину серед усіх маршрутів, які їх сполучають.

Будемо вважати, що усі ребра графа мають однакову довжину, рівну одиниці, тобто усі ребра рівноправні. Такий граф називають **ненавантаженим.**

За допомогою ненавантажених графів описують системи, які можуть переходити з одного стану в інший. Усі стани рівноправні (однаково ймовірні).

Алгоритм Беллмана

Кожній вершині графа приписують індекс так, що він дорівнює довжині найкоротшого маршруту, який сполучає цю вершину з кінцевою V_k.

1. Індексування вершин графа починають з кінцевої вершини V_k. Їй приписують індекс 0. Усім суміжним вершинам з V_k приписують індекс 1.

2. Якщо вершині V_i приписано індекс λ_i, то суміжним з нею вершинам, яким ще не приписано індекс, приписують індекс $\lambda_{i+1} = \lambda_i + 1$. Якщо одній і тій же вершині можна приписати декілька індексів, то вибирають найменший.

3. Процес приписування індексів продовжують до тих пір, поки усі вершини не будуть проіндексовані. Пошук мінімального маршруту починають із початкової вершини V_p, рухаючись до тієї вершини в якої менший індекс.

Відшукання мінімального маршруту у навантаженому графі

Нехай задано граф $G = (U, V)$ і кожному йому ребру V_iV_j приписана довжина $l(V_iV_j)$, яка називається **вагою** ребра.

Потрібно відшукати маршрут мінімальної довжини, який з'єднує вершини V_p і V_k. Алгоритм Белманна для розв'язання цієї задачі не застосовний, бо може існувати маршрут меншої довжини, але який складається з більшої кількості ребер. Тому для відшукання мінімального маршруту у навантажених графах користуються наступним алгоритмом.

1. Кінцевій вершині приписують індекс 0, усім іншим ∞.

2. Для вершини V_j шукають те ребро, для якого виконується умова $\lambda_j - \lambda_i \geq l(V_i V_j)$ (різниця індексів більш рівна довжині ребра). Вершині V_j приписують індекс $\lambda_j = \lambda_i + l(V_i V_j)$. Процес приписування індексів продовжують до тих пір, поки можна зменшити індекс вершини V_j.

3. Коли усі вершини будуть проіндексовані, починають пошук мінімального маршруту з початкової вершини. Для кожної вершини шукають те ребро, для якого різниця індексів дорівнює довжині ребра. Процес буде скінченним, оскільки на кожному кроці значення індексу зменшується на довжину ребра.