PRIM ALGORITMI

Onarkulov Maksadjon Karimberdiyevich
Farg'ona davlat universiteti Amaliy matematika va informatika kafedrasi dotsenti (PhD)
maxmaqsad@gmail.com
Qirg'izboyev Diyorbek Akmaljon o'g'li
Farg'ona davlat universiteti 2-bosqich talabasi
diyorbekqirg'izboyev91@gmail.com

Annotatsiya:

Prim algoritmi graflar nazariyasida minimal kapsayıcı ağaç (Minimum Spanning Tree - MST) topish uchun ishlatiladigan eng samarali algoritmlardan biridir. U 1957-yilda Robert Prim tomonidan ishlab chiqilgan bo'lib, Jarník-Prim algoritmi nomi bilan ham tanilgan. Prim algoritmi og'irlikli bog'langan grafda barcha tugunlarni bog'laydigan eng kichik umumiy og'irlikka ega bo'lgan ostov daraxtni topadi. Algoritm ochko'z (greedy) yondashuvga asoslangan bo'lib, har bir qadamda eng kichik og'irlikka ega qirralarni tanlaydi. Ushbu maqolada Prim algoritmining ishlash prinsipi, asosiy bosqichlari, murakkablik tahlili, boshqa MST algoritmlari bilan taqqoslash va amaliy qo'llanilishi haqida batafsil ma'lumot beriladi.

Kalit so'zlar: *Prim algoritmi, minimal kapsayıcı ağaç, MST, graflar nazariyasi, ochko'z algoritm, Jarník-Prim, bog'langan graf, ostov daraxt, og'irlikli graf, optimizatsiya, dasturlash, algoritm tahlili*

Annotation:

Prim's algorithm is one of the most efficient algorithms for finding Minimum Spanning Trees (MST) in graph theory. Developed by Robert Prim in 1957, it is also known as the Jarník-Prim algorithm. Prim's algorithm finds a spanning tree with minimum total weight that connects all vertices in a weighted connected graph. The algorithm is based on a greedy approach, selecting the edge with minimum weight at each step. This paper provides detailed information about

Prim's algorithm working principle, main stages, complexity analysis, comparison with other MST algorithms, and practical applications.

Keywords: *Prim's algorithm, minimum spanning tree, MST, graph theory, greedy algorithm, Jarník-Prim, connected graph, spanning tree, weighted graph, optimization, programming, algorithm analysis*

Аннотация:

Алгоритм Прима является одним из наиболее эффективных алгоритмов для поиска минимального остовного дерева (MST) в теории графов. Разработанный Робертом Примом в 1957 году, он также известен как алгоритм Ярника-Прима. Алгоритм Прима находит остовное дерево с минимальным общим весом, соединяющее все вершины во взвешенном связном графе. Алгоритм основан на жадном подходе, выбирая ребро с минимальным весом на каждом шаге. В данной статье представлена подробная информация о принципе работы алгоритма Прима, основных этапах, анализе сложности, сравнении с другими алгоритмами MST и практическом применении.

Ключевые слова: *алгоритм Прима, минимальное остовное дерево, MST, теория графов, жадный алгоритм, Ярник-Прим, связный граф, остовное дерево, взвешенный граф, оптимизация, программирование, анализ алгоритма*

Zamonaviy kompyuter fanlari va matematikasida graflar nazariyasi muhim o'rin tutadi. Graflar yordamida real hayotdagi murakkab tizimlarni modellashtirish va ular ustida turli optimizatsiya masalalarini yechish mumkin. Bunday masalalardan biri minimal kapsayıcı ağaç (Minimum Spanning Tree - MST) topish hisoblanadi. MST masalasi transport tarmoqlari, telekommunikatsiya tizimlari, kompyuter tarmoqlari va boshqa ko'plab sohalarda qo'llaniladi.

Minimal kapsayıcı ağaç - bu bog'langan og'irlikli grafning barcha tugunlarini bog'laydigan, lekin hech qanday tsikl hosil qilmaydigan qirralarning eng kichik

umumiy og'irlikka ega to'plamidir. MST topish uchun bir nechta algoritm ishlab chiqilgan bo'lib, ulardan eng mashhurlari Prim, Kruskal va Borůvka algoritmlaridir.

Prim algoritmi 1930-yilda chex matematigi Vojtěch Jarník tomonidan birinchi marta taklif qilingan, keyinchalik 1957-yilda amerikalik kompyuter olimi Robert Prim tomonidan qayta kashf etilgan va rivojlantirilgan. Shuning uchun bu algoritm Jarník-Prim algoritmi nomi bilan ham tanilgan. Dijkstra algoritmi bilan o'xshash strukturaga ega bo'lgan Prim algoritmi ochko'z yondashuvga asoslangan bo'lib, har bir bosqichda eng optimal tanlov qiladi.

Prim algoritmining asosiy g'oyasi oddiy: biror tugundan boshlab, har safar MST ga qo'shilmagan tugunlar orasidan eng kichik og'irlikka ega qirra orqali erishish mumkin bo'lgan tugunni tanlash. Bu jarayon barcha tugunlar MST ga qo'shilgunga qadar davom etadi. Algoritmning bu yondashuvi uni amaliy ilovalar uchun juda qulay qiladi, chunki u oddiy implementatsiyaga ega va ko'plab real masalalarda samarali ishlaydi.

Hozirgi kunda Prim algoritmi nafaqat nazariy jihatdan, balki amaliy dasturlashda ham keng qo'llaniladi. Tarmoq dizayni, klaster tahlili, kompyuter grafikasi va sun'iy intellekt sohalarida bu algoritm muhim rol o'ynaydi. Ushbu maqolada Prim algoritmining to'liq tahlili, uning ishlash mexanizmi, afzalliklari va kamchiliklari, shuningdek, boshqa algoritmlar bilan taqqoslash beriladi.

Prim algoritmining ishlash prinsipi

Prim algoritmi minimal kapsayıcı ağaç topish uchun ochko'z (greedy) strategiyadan foydalanadi. Algoritmning asosiy prinsipi shundan iboratki, har bir qadamda mavjud MST ga eng kichik og'irlikka ega qirra orqali yangi tugun qo'shiladi. Bu jarayon barcha tugunlar MST ga kiritilgunga qadar takrorlanadi.

Algoritmning ishlash bosqichlari quyidagicha:

Boshlang'ich bosqich: Ixtiyoriy bir tugunni tanlash va uni MST ning dastlabki tuguni sifatida belgilash. Bu tugun "kirgan tugunlar" to'plamiga qo'shiladi, qolgan barcha tugunlar esa "kirmagun tugunlar" to'plamida qoladi.

Asosiy sikl: Har bir iteratsiyada "kirgan tugunlar" dan "kirmagan tugunlar" ga yo'nalgan qirralar orasidan eng kichik og'irlikka ega bo'lganini tanlash. Tanlangan qirra MST ga qo'shiladi va uning oxirgi tuguni "kirgan tugunlar" to'plamiga o'tkaziladi.

Tugash sharti: Barcha tugunlar "kirgan tugunlar" to'plamiga kirganda algoritm to'xtaydi. Natijada hosil bo'lgan qirralar to'plami minimal kapsayıcı ağaçni tashkil qiladi.

Prim algoritmining to'g'riligini isbotlash uchun ochko'z tanlovning optimalligini ko'rsatish kerak. Har bir qadamda tanlangan qirra haqiqatan ham MST ning bir qismi bo'lishi kerak. Bu cut property (kesish xossasi) yordamida isbotlanady: agar graf ikki qismga bo'linsa va ular orasidagi eng kichik og'irlikka ega qirra MST ning bir qismi bo'ladi.

Algoritmning ishlashi davomida prioritet navbati (priority queue) ma'lumotlar strukturasidan foydalaniladi. Bu struktura har doim eng kichik og'irlikka ega qirrani tezda topish imkonini beradi. Prioritet navbati sifatida odatda min-heap ishlatiladi, bu esa algoritmning umumiy murakkabligini kamaytiradi.

Prim algoritmi Dijkstra algoritmi bilan ko'p o'xshashlikka ega, lekin maqsadi boshqa. Dijkstra eng qisqa yo'l topsa, Prim minimal kapsayıcı ağaç topadi. Ikkala algoritm ham bir xil ma'lumotlar strukturasidan foydalanadi va o'xshash mantiqqa ega, lekin har bir qadamda turli mezonlar bo'yicha tanlov qiladi.

Prim algoritmining afzalliklari va kamchiliklari

Prim algoritmi bir qator muhim afzalliklarga ega bo'lib, bu uni ko'plab amaliy masalalarda qo'llash uchun mos qiladi. **Asosiy afzalliklaridan** biri - algoritmning soddaligi va tushunarligi. Uning ishlash prinsipi intuitive bo'lib, dasturlash va debug qilish oson. Shuningdek, Prim algoritmi **barqaror (stable)** ishlaydi, ya'ni har doim to'g'ri natija beradi, agar graf bog'langan bo'lsa.

Prim algoritmining yana bir muhim afzalligi - u **zich (dense) graflar** uchun juda samarali. Graf qirralari soni tugunlar sonining kvadratiga yaqin bo'lgan holatlarda Prim algoritmi Kruskal algoritmidan tezroq ishlaydi. Bu ayniqsa to'liq

yoki deyarli to'liq graflar bilan ishlashda muhim. Algoritmning **vaqt murakkabligi** ma'lumotlar strukturasiga bog'liq: oddiy implementatsiyada $O(V^2)$, binary heap bilan $O((V+E)\log V)$, Fibonacci heap bilan esa $O(E + V \log V)$ ni tashkil qiladi.

Prim algoritmi **onlayn (incremental)** ishlay oladi, ya'ni yangi tugun qo'shilganda butun MST ni qaytadan hisoblash shart emas. Bu xususiyat dinamik graflar bilan ishlashda foydali. Shuningdek, algoritm MST ning qurilish jarayonini vizuallashtirish uchun ham qulay, chunki har bir qadamda yangi qirra qo'shiladi.

Biroq Prim algoritmining **ayrim kamchiliklari** ham mavjud. Birinchi navbatda, u **siyrak (sparse) graflar** uchun Kruskal algoritmiga nisbatan sekinroq ishlashi mumkin. Qirralar soni tugunlar soniga nisbatan kam bo'lgan graflar uchun Kruskal ko'pincha samaraliroq. Shuningdek, Prim algoritmi **qo'shimcha xotira** talab qiladi, ayniqsa prioritet navbati uchun.

Yana bir kamchilik - algoritm **parallel implementatsiya** uchun qiyin. Chunki har bir qadam oldingi qadamning natijasiga bog'liq bo'lib, parallellashtirish murakkab masala. Kruskal algoritmi bu jihatdan ko'ra osonroq parallellashtirila oladi. Shuningdek, **floating-point** og'irliklar bilan ishlashda aniqlik masalalari paydo bo'lishi mumkin. Umuman olganda, Prim algoritmi o'zining soddaligi, ishonchliligi va zich graflar uchun samaradorligi bilan ajralib turadi. Uning kamchiliklari ko'pincha muayyan kontekstda muhim bo'lib, to'g'ri implementatsiya va ma'lumotlar strukturasini tanlash orqali ko'plab muammolardan qochish mumkin. Real ilovalarda Prim algoritmi o'zining afzalliklari tufayli keng qo'llaniladi.

Prim algoritmining amaliy qo'llanilishi

Prim algoritmi haqiqiy hayotning ko'plab sohalarida amaliy qo'llaniladi. Uning eng keng tarqalgan qo'llanilish sohalaridan biri **tarmoq dizayni**dir. Telekommunikatsiya kompaniyalari shaharlar orasida eng kam xarajat bilan kabel tarmoqlarini qurish uchun Prim algoritmidan foydalanadilar. Bu yerda tugunlar shaharlarni, qirralar esa ular orasidagi kabel uzunligi yoki qurish xarajatlarini

ifodalaydi. Algoritm barcha shaharlarni bog'laydigan eng arzon tarmoq topishga yordam beradi.

Transport tizimlarida ham Prim algoritmi muhim rol o'ynaydi. Yo'l qurilishi loyihalarida, temir yo'l tarmoqlarini rejalashtirish va hatto havo yo'llarining marshrutlarini optimallashtirish uchun ishlatiladi. Masalan, yangi metro liniyasini loyihalashda har bir bekatni bog'laydigan eng qisqa tunnel tarmog'ini topish uchun Prim algoritmi qo'llaniladi.

Kompyuter tarmoqlari sohasida Prim algoritmi tarmoq topologiyasini optimallashtirish uchun ishlatiladi. LAN tarmoqlarini qurish, internet provayderlarining infrastrukturasini rejalashtirish va klaster hisoblash tizimlarida tugunlar orasidagi bog'lanishlarni optimallashtirish uchun qo'llaniladi. Bu yerda maqsad - barcha kompyuterlarni bog'laydigan eng kam xarajatli yoki eng tez ishlaydigan tarmoqni yaratish.

Kompyuter grafikasida Prim algoritmi mesh segmentatsiyasi va 3D modellarni soddalashtirish uchun ishlatiladi. Shuningdek, **ma'lumotlarni klasterlashtirish** (clustering) algoritmlarida ham Prim dan foydalaniladi. Bu yerda o'xshash ma'lumotlar nuqtalari orasidagi masofani hisobga olgan holda eng optimal klasterlar hosil qilinadi.

Biologiya va tibbiyot sohalarida Prim algoritmi filogenetik daraxtlar qurish uchun ishlatiladi. Turlarning evolutsion rivojlanishini o'rganishda genetik masofalarni hisobga olgan holda eng ehtimoliy filogenetik daraxt topiladi. Shuningdek, **neyron tarmoqlari**ni optimallash va **molekulyar biologiya**da protein strukturalarini tahlil qilishda ham qo'llaniladi.

Elektr energetikasida quvvat tarmoqlarini loyihalash uchun Prim algoritmi ishlatiladi. Elektr stansiyalarini iste'molchilarga bog'laydigan eng samarali uzatish liniyalari tarmog'ini topish maqsadida qo'llaniladi. **Suv ta'minoti tizimlari**ni loyihalash va **gaz quvurlari** tarmog'ini optimallashtirish uchun ham foydalaniladi.

Zamonaviy **IoT (Internet of Things) tizimlari**da Prim algoritmi sensor tarmoqlarini optimallashtirish uchun ishlatiladi. Bu yerda maqsad - barcha

sensorlarni bog'laydigan eng kam energiya sarflaydigan aloqa tarmog'ini yaratish. **Robotika**da esa harakatlanish yo'llarini rejalashtirish va ko'p robotli tizimlarni koordinatsiya qilish uchun qo'llaniladi.

Xulosa

Prim algoritmi graflar nazariyasining eng muhim va amaliy algoritmlaridan biri bo'lib, minimal kapsayıcı ağaç topish masalasini samarali hal qiladi. Uning oddiy, tushunarli strukturasi va ishonchli ishlashi uni ko'plab real hayot masalalarida qo'llash uchun ideal qiladi. Algoritmning ochko'z yondashuvi har doim optimal natija berishini kafolatlaydi, bu esa uni nazariy jihatdan ham, amaliy jihatdan ham qimmatli qiladi. Prim algoritmining zich graflar uchun yuqori samaradorligi, onlayn ishlash qobiliyati va visualization uchun qulayligi uni boshqa MST algoritmlaridan ajratib turadi. Garchi siyrak graflar uchun Kruskal algoritmi afzalroq bo'lsa-da, ko'plab amaliy vaziyatlarda Prim algoritmi optimal tanlov hisoblanadi. Uning telekommunikatsiya, transport, kompyuter tarmoqlari va boshqa sohalardagi keng qo'llanilishi uning amaliy qiymatini isbotlaydi.

Kelajakda Prim algoritmini yanada takomillashtirish yo'llari mavjud. Parallel va taqsimlangan implementatsiyalar, yangi ma'lumotlar strukturalari bilan integratsiya va yangi qo'llanilish sohalarini o'rganish dolzarb tadqiqot yo'nalishlari hisoblanadi. Shuningdek, katta hajmdagi graflar (big graphs) va real vaqt rejimida ishlaydigan tizimlar uchun algoritmni optimallash ham muhim vazifa. Dasturchilar va algoritm tahlilchilari uchun Prim algoritmi nafaqat muayyan masalani yechish vositasi, balki ochko'z algoritmlar va graflar nazariyasini chuqur tushunish uchun muhim ta'lim vositasi ham hisoblanadi. Uning to'g'riligi isboti, murakkablik tahlili va amaliy implementatsiyasi algoritmik tafakkurni rivojlantirishda muhim rol o'ynaydi.

Foydalanilgan adabiyotlar

1. Prim, R.C. *Shortest connection networks and some generalizations*. Bell System Technical Journal, 1957.

2. Thomas H. Cormen et al. *Introduction to Algorithms*. MIT Press, 4th Edition, 2022.

3. Robert Sedgewick, Kevin Wayne. *Algorithms*. Addison-Wesley, 4th Edition, 2011.

4. Jarník, V. *O jistém problému minimálním*. Práce Moravské Přírodovědné Společnosti, 1930.

5. Fredman, M.L., Tarjan, R.E. *Fibonacci heaps and their uses in improved network optimization algorithms*. Journal of the ACM, 1987.

6. Nesetril, J., Milková, E., Nesetrilová, H. *Otakar Borůvka on minimum spanning tree problem*. Discrete Mathematics, 2001.

7. Graham, R.L., Hell, P. *On the history of the minimum spanning tree problem*. Annals of the History of Computing, 1985.

8. Chazelle, B. *A minimum spanning tree algorithm with inverse-Ackermann type complexity*. Journal of the ACM, 2000.

9. Gabow, H.N., Galil, Z., Spencer, T., Tarjan, R.E. *Efficient algorithms for finding minimum spanning trees in undirected and directed graphs*. Combinatorica, 1986.

10. Wikipedia. *Prim's algorithm*. https://en.wikipedia.org/wiki/Prim%27s_algorithm

11. GeeksforGeeks. *Prim's Minimum Spanning Tree Algorithm*. https://www.geeksforgeeks.org

12. Skiena, S.S. *The Algorithm Design Manual*. Springer, 3rd Edition, 2020.

13. Kleinberg, J., Tardos, E. *Algorithm Design*. Pearson, 2005.

14. Dasgupta, S., Papadimitriou, C., Vazirani, U. *Algorithms*. McGraw-Hill, 2006.

15. Lecture Notes on Graph Algorithms. *Minimum Spanning Trees*. Computer Science Department, Stanford University.

Ilm fan taraqqiyotida raqamli iqtisodiyot va zamonaviy ta'limning o'rni hamda rivojlanish omillari