Network Optimization by Matrix Multiplication

김주영 / 2019.03.31.

Table of Contents

1. Introduction

블록체인의 상용화가 진전됨에 따라 네트워크 부하도 증가하고 있다. 이에 일반적으로 사용되고 있는 인터넷 연결망을 행렬로 표현한 다음, 최소한의 비용으로 통신하는 방법을 결정하는 알고리즘에 대해 살펴볼 것이다.

2. Modifying centralized network into matrices

인터넷 연결망에서 각 노드들의 연결 상태를 표현하기 위해 아주 전형적인 자료구조인 인접행렬을 사용하고자 한다. 이는 노드들이 여러 개의 인터넷 기지국과 동시에 연결하여 병렬 통신을 하거나 인터넷 두절 시에 다른 인터넷 기지국에 연결하는 등의 최근의 인터넷 행태를 반영하기 위해서는 이보다 빠르게 연산할 수 있는 자료구조를 사용하기 어렵기 때문이다.

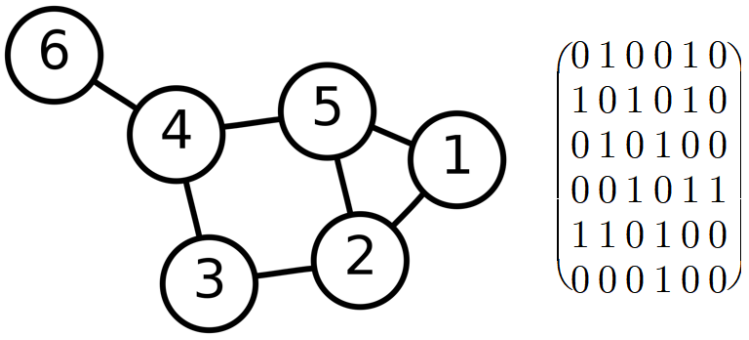

Figure 1. 간단한 인터넷 연결망과 인접행렬 표현. 2와 4는 기지국이다.

3. Iterative deepening uniform cost search

실제 인터넷 연결망에서는 모든 구간 통신마다 서로 다른 시간 비용이 발생하며 가장 효율적인 통신을 위해서는 시간 비용을 최소화해야 할 것이다. 블록체인 통신에서는 어떤 노드와 그 노드에서 가장 가까운 노드들끼리 통신하면 되므로, 그래프에서 특정 노드에서 가장 적은 비용으로 도달할 수 있는 순으로 노드를 정렬해 주는 균일 비용 탐색을 사용할 수 있다.

그러나 이는 모든 시간 비용이 서로 다른 상황일 때를 가정한 것으로, 어떤 노드들 간의 통신 비용이 다른 어떤 노드들 간의 통신 비용과 같을 수 있는 실제 인터넷 연결망에서는 균일 비용 탐색에 순차 증가 방식을 추가적으로 도입하는 것이 필요하다.

이렇게 하면 특정 노드에서 새로운 트랜잭션에 대한 정보를 보낼 노드들의 우선순위를 반드시 결정할 수 있다. 특정 노드들 간의 통신이 단방향 통신이어도 관계 없다.

4. Finding out whether a node is reachable by matrix multiplication

그런데 특정 노드들 간에는 단방향 통신만 존재하여 실질적으로 트랜잭션에 대한 정보를 보낼 수 없는 경우가 있다. 이를 파악하기 위해 인접 행렬을 거듭제곱할 수 있다. 인접행렬을 k제곱하면 특정 노드에서 다른 특정 노드까지 k-1개의 다른 노드를 거쳐서 갈 수 있는 방법의 가짓수를 담은 새로운 행렬이 만들어진다. 그래서 만일 인접행렬을 n-1제곱했는데도 이용하고자 하는 통신 경로의 값이 계속 0인 경우 해당 경로는 절대로 이용할 수 없다.

어떤 노드가 완전히 고립된 경우에는 인접행렬을 기약 행 사다리꼴(RREF)로 만들었을 때 모든 원소가 0인 열이 만들어지는데, 이 경우에는 인접행렬의 역행렬이 존재하지 않으며, 아래에서 설명할 위수도 존재하지 않기 때문에 행 축약을 하지 않는다면 정말로 인접행렬을 n-1제곱해야 알 수 있다. 하지만 어떤 노드가 고립되었다면 그 노드의 존재가 네트워크에 전파되지 않았을 것이므로 고립된 노드가 있는 그래프는 존재할 수 없다. 하지만 유의할 것은, 고립되지 않은 노드라고 해서 통신이 가능한 것은 아니라는 점이다. 인접 행렬 상에서 수신만 가능하고 발신이 불가능한 노드가 존재할 수 있기 때문이다.

그리고 인접행렬을 k제곱했을 때 자기 자신을 거쳐서 가는 의미 없는 경로를 세지 않도록 인접행렬의 대각성분의 값은 모두 0으로 정해야 한다.

5. Comparing effectiveness of graph search and matrix multiplication

위에서 밝힌 순차 증가 균일 비용 탐색의 시간복잡도는 $O(b^{1+c/e})$이고, 행렬곱을 k회 반복할 때의 시간복잡도는 간단하게 알 수 있듯이 $O(kn^3)$이다. 물론 그래프 탐색 알고리즘을 통해서도 도달 여부를 판단할 수 있지만, 양쪽의 최악의 경우에는 k=n인 대신 b=n이고 c/e가 3 이상이 되는데 이렇게 되면 통념과는 달리 행렬곱이 그래프 탐색보다 훨씬 빠르다.

균일 비용 탐색의 시간복잡도에 사용된 변수들에 대한 구체적인 정의는 다음과 같다.

- b: 탐색 과정에서 한 노드가 연결된 다른 노드들의 수
- c: 그래프에서 가장 비용이 높은 경로의 비용
- e: 그래프에서 가장 비용이 낮은 간선의 비용

6. Matrix order and general linear group

행렬에 대해서 말하자면 일반선형군 $GL(n, Z_p)$는 행렬의 모든 원소가 0 이상 p 미만인 정수이며 역행렬이 존재하는 n차 정사각행렬만을 원소로 가지는 군이다. 그리고 행렬의 특성다항식은 그 행렬을 대입했을 때의 결과가 영행렬이 되면서 최고차항의 계수가 1이고 최고차항의 차수가 최소인 행렬의 다항식을 말한다. 인접행렬은 처음에는 모두 $GL(n, Z_2)$에 속한다.

특성다항식의 위수는 그 특성다항식이 x^k-1의 인수가 되는 최소의 k의 약수이다. 행렬의 위수는 행렬을 k제곱하면 단위행렬이 되는 최소의 k이다. 이때 행렬의 위수는 그 행렬의 특성다항식의 위수의 약수이다. 따라서 행렬의 특성다항식을 구하면 행렬의 위수가 될 수 있는 후보들과 위수의 값의 상한선을 알 수 있다. 그러므로 행렬을 n-1제곱해보지 않아도 된다.

그런데 일반선형군은 곱셈에 대해 닫혀 있기 때문에, $GL(n, Z_p)$에 속하는 행렬을 k제곱한다는 것은 행렬곱에 의해 얻은 행렬을 그대로 구하는 것이 아니라, 만일 행렬곱의 결과의 원소 중에 값이 p 이상인 것이 있으면 mod p 연산을 해서 다시 $GL(n, Z_p)$에 속하게 한다.

하지만 인접 행렬에 대해서 바로 mod p 연산을 할 수는 없다. 그것은 특정 노드에서 k개의 노드를 거쳐 특정 노드로 갈 수 있는 경로의 수가 p의 배수 개가 될 수 있기 때문이다. 그럼에도 불구하고 실제 인터넷 접속 과정과 같이 중앙 집중적인 그래프인 경우 인접행렬의 원소로 1보다는 0이 많기 때문에 인접행렬의 k제곱은 자연스럽게 다시 $GL(n, Z_p)$에 속하게 된다.

이것이 중요한 이유는 만일 그렇게 되지 않으면 인접행렬의 k제곱은 최대 다음의 값을 원소로 포함하는 행렬이 되기 때문이며, 이는 n-1에 비해 매우 큰 값이므로 실용적이지 않다.

$$MAX = n^{2^k - 1}(p-1)^{2^k}$$

Figure 2. 행렬곱을 통해 얻어지는 행렬의 원소의 최댓값. n-1에 비해 매우 크다.

7. Lagrange Theorem

라그랑주의 정리에 따르면 $GL(n, Z_p)$에 속하는 행렬의 위수는 $|GL(n, Z_p)|$의 약수이며, 이와는 별개로 $GL(n, Z_p)$에 속하는 행렬의 위수는 p^n-1 이하로, $|GL(n, Z_p)|$에 비해 매우 작다. 이러한 점을 활용한다면 행렬의 위수의 후보로 $|GL(n, Z_p)|$의 작은 약수들을 거론할 수 있기에 심지어 특성다항식을 구하지 않고도 행렬의 위수의 후보를 쉽게 알 수 있다. 그리고 당연한 얘기지만 인접행렬은 최대 n-1번 거듭제곱할 것이기 때문에 이들 중에 n-1 이상인 것은 버리고 n-1보다 작은 것을 취해서 그 후보들만큼 거듭제곱했을 때마다 원래의 인접행렬과 같은지 확인하면 된다.

만일 운이 좋게 인접행렬의 위수가 매우 작고 인접행렬을 그 위수까지 제곱했을 때 다시 $GL(n, Z_p)$에 속한다면 지금까지의 연결 상태만을 가지고 특정 노드에서 다른 특정 노드에 도달할 수 있는지를 판단할 수 있다.

위수	1	2	3	4	7	계
행렬의 수	1	21	56	42	48	168

Table 1. $GL(3, Z_2)$의 원소의 위수 분류. $|GL(3, Z_2)|$=168이다.

8. Conclusion

행렬곱은 가장 적은 비용으로 도달할 수 있는 노드를 알려주지는 않기 때문에 어떤 노드를 방문할 수 있는지를 파악하기 위해서만 사용할 수 있다. 그렇지만 인터넷에 접속하기 위해서는 일반적으로 교환국 등 최소한 3개 이상의 노드를 거치기 때문에 행렬곱은 여전히 유용하다.

문제는 내선망과 같이 직접적인 연결이 많은 경우 c/e가 3 이상인지를 그래프 탐색을 하기 전에는 알 수 없다는 것이다. 그리고 블록체인 네트워크는 점점 커지기 때문에 일단 방법을 결정하면 되돌리기 위해 많은 시간 비용이 발생하게 된다. 이를 해소하기 위해 그래프의 임의 지점 몇 개를 선정한 다음, 간선이 적어서 직접적인 연결이 적을 것으로 예상되고, 주위의 간선의 비용을 측정했을 때 이들의 최고 비용과 최소 비용의 비가 커서 c/e가 높을 것으로 예상되는 경우에는 먼저 행렬곱을 통해 해당 노드와 통신이 가능한지 확인해 보고, 그렇지 않으면 바로 균일 비용 탐색을 시도할 수 있다. 또는 가장 비용이 높은 구간의 비용을 예측하는 휴리스틱을 사용하여 빠르게 결정할 수도 있으나, 이는 노드 간의 직선 거리를 휴리스틱으로 사용하는 경우를 제외하면 상당히 부정확하므로 통신할 수 있는 노드가 많은 경우에 사용해볼 수 있다.

References

Gabe Cunningham (2005). "The General Linear Group". http://www-math.mit.edu/~dav/genlin.pdf (검색일: 2019.03.31.).

강수민 · 김주영 · 박성우 (2016). "GL(n, Z_p)에서 원소의 위수의 최댓값에 대한 연구". 창원과학고등학교 과제연구논문. http://nestian.kr/documents/matrix.pdf (검색일: 2019.03.31.).