Network Flow

14004 구재현

What’s Network Flow?

- s - t간의 simple path를 구하는 것도 의미 있지만
- s - t간에 일정한 양을 여러 path에 대해서 흘려주는 것도 의미있을 수도
- 실제로 의미있음. Network Flow는 “용량” 제한이 걸려있는 그래프에서 최대로 얼마만큼 흘려줄 수 있는지를 계산해줌
- 플로우 그 자체보다는 응용할 수 있는 곳이 많아서 의미있음
- 그래도 모르겠으면 http://koistudy.net/?mid=prob_page&NO=667

Disclaimer

- 1. 일단 본인도 이 분야에 대해서 별로 전문적이지 않아서 증명은 대부분 exercise to the viewer로.. 네트워크 플로우는 자명하지 않은 정리들이 무지 많이 나옴. 쉽게 증명되지 않는 것들도 많으니 해답은 인터넷이나 책을 참고
- 2. Network Flow / 이분 매칭 정도는 고등학교 대회에서 간혹.. 보이지만, 그 이상은 거의 대학생 레벨임. 아주 어렵진 않지만 기반 지식이 수준이 높은 편이라.. 편히 들으셈.

Table of Contents

Network Flow

- http://koistudy.net/?mid=prob_page&NO=667
- 대강 무슨 문제인지 알게 될듯 (...)
- 이러한 문제는 Ford-Fulkerson Method (Edmonds-Karp Algorithm) 으로 해결 가능
- 간단히 말하면, 아무 경로나 잡아서 죽어라 흘려보는 알고리즘 (...)

Ford-Fulkerson Method

- 1. 말 그대로 capacity > 0인 아무 경로나 잡는다 (BFS, DFS)
- 2. 이 때 흘려준 값은 경로에 있는 Capacity의 최솟값
- 3. 경로상에서 Capacity를 흘려준 값만큼 빼주고, 역변에 더해준다
- 4. 그러한 경로가 없을 때까지 이를 반복
- Q1. 이 알고리즘은 정당한가?
- 복잡도는? f = Maxflow의 값일때, DFS를 사용하면 O(f(V+E)), BFS를 사용하면 O(Min(VE,f) * (V+E)
- Q2. 왜 저러한 복잡도가 나오는가?

Min-cut Max-flow Thm

- 두 정점 S - T가 있으며 Weighted Edge가 있다.
- 두 정점 S - T를 떼어놓고 싶다! 떼어놓는 비용은 에지의 가중치합. 최소 비용은?
- = Mininum Cut Problem.
- Maximum Flow와 같은 문제?!
- Q3 : Prove that Min-Cut == Max-Flow.
- Q4 : How to Track Min-Cut Edge?
- http://koistudy.net/?mid=prob_page&NO=1236 (<- Min-Cut)
- http://koistudy.net/?mid=prob_page&NO=2006 (<- 정점 버전?!)

Dinic’s Algorithm

- Edmonds-Karp Algorithm (BFS + Ford-Fulkerson) 의 시간 복잡도는 O(VE^2).
- Dinic’s Algorithm은 O(V^2E)에 네트워크 플로우를 구할 수 있다.
- 찾아보세요!

Bipartite Matching

- Bipartite Graph (이분 그래프)
- 정의 : 상호간에 에지가 없는 두 집합으로 나눌 수 있는 그래프.
- 최소 채색수가 2인 그래프.
- Matching (매칭)
- 간선의 부분집합인데, 간선 간에 겹치는 정점이 없는 부분집합.
- Maximum Matching is O(V^2E), But It's VERY HARD TO CODE.
- Bipartite Matching is O(VE) Using Network Flow.

Bipartite Matching

- START
- (AAAAAAAAAAAAAAAAAAAAAA)   (<- A번 색깔)
- XXXXXXXXXIXIXIXIXIXXXIXXIXII   (<- 에지)
- (BBBBBBBBBBBBBBBBBBBBBB)   (<- B번 색깔)
- END
- START -> END 간의 Max-Flow값이 Maximum Matching.
- Q5. Prove that Max-Flow == Max-Match in Bipartite Graph.
- Q6. How to print matched edges?

Bipartite Matching

- 이분 매칭은 BFS를 써도, DFS를 써도 됩니다.
- (심지어 DFS 쓰는게 더 빠릅니다 ㅋㅋ)
- O(fE) 라는 식을 상기시켜보면, f = O(V) 이기 때문에, 시간 복잡도가 O(VE). 실제론 이보다도 빠르게 돌아감.
- O(Sqrt(V)E)에 이분 매칭을 구하는 Hopcroft-Karp Algorithm이 존재함. 배워보세요!

Vertex Cover, Indep Set

- What is Minimum Vertex Cover?
- 정점의 부분집합으로써, 모든 간선 E = (u,v)에 대해 u나 v가 집합에 속해 있는 관계를 만족하면 Vertex Cover라고 함. (u || v)
- What is Maximum Independent Set?
- 정점의 부분집합으로써, 모든 간선 E = (u,v)에 대해 u와 v가 모두 집합에 속해있는 관계를 만족하지 않으면 Independent Set이라 함. !(u && v)
- Vertex Cover의 여집합은 Independent Set.
- Both are NP-hard in General, O(VE) in Bipartite Graph

Minimum Vertex Cover

- 이분 그래프에 대해서, Minimum Vertex Cover == Maximum Matching
- Konig's Thm.
- Q7. Prove Konig's Thm.
- Q8. print MVC / MIS in Bipartite Graph

MCMF

- Min-cost Flow. Min-cost Max-flow
- 그래프의 Edge가, Capacity와 Cost를 가진다면?
- Flow의 양은 정해져 있다 -> 난 이만큼 흘리고 싶다!
- 기준은 마음대로임, Max Flow만큼 흘리고 싶을 수도 있고..
- 최소 비용으로 흘리는 방법!

How to solve MCMF?

- Max-Flow를 찾는다... 당연한가?
- BFS와 DFS를 사용하지 않고, 최단경로 알고리즘을 사용해서 (다익스트라 등) 흘려줄 경로를 찾는다.
- 최단 경로는 Cost에만 dependent. cap은 0보다 크기만 하면 상관없음.
- 역변 처리는? 역변의 cost는 정변의 cost의 음수값이다. 참 쉽죠?
- Q9. Prove there is no negative cost cycle in the graph.
- Q10. Prove MCMF algorithm.
- 음수 사이클은 없는데, 음수 간선은 처리할 수 있는 최단경로 알고리즘이 존재하면, 문제는 해결됨.

Bellman-Ford / SPFA

- 다익스트라 알고리즘은 음수 간선이 나오면 터진다 다익스트라니뮤ㅠㅠㅠㅠ
- 벨만 포드는 O(VE)에 동작하는 최단경로 알고리즘, 사이클만 없으면 음수 간선에도 끄떡없음
- 벨만 포드를 약간 확장한 Shortest Path Faster Algorithm (SPFA)가 있음. 큐를 사용해서 커팅해 주는데 준 선형에 돌아감. 심지어 학자들도 왜 빠른지 몰라..

시간 복잡도 분석

- 일단 O(fVE) 이다.
- flow가 unbounded면 어떡하지? ㅠㅠ (저도 잘 모름)
- bounded된 경우만 생각해보자. MCMF로 무슨 문제를 풀 수 있나?
- Minimum Sum (Large)!
- Q11 : Solve Minimum Sum (Large) in O(n^4).