알고리즘의 기초 HW #3

전기정보공학부 2016-10769 이권형

1. a_i와 b_i를 모두 non-decreasing order로 정렬하면 payoff가 최대가 된다. 증명은 greedy algorithm을 사용하여 다음과 같다.

위 algorithm이 optimal하지 않다고 가정하고, $a_1 \le a_k$인 k에 대해, non decreasing order로 정렬된 S'와 a_1과 a_k를 바꾼 S가 있다고 하고, S'이 optimal이라 가정한다. 이때, $\frac{S_{pay}}{S'_{pay}} = \frac{\prod_S a_i^{b_i}}{\prod_{S'} a_i^{b_i}} = \left(\frac{a_1}{a_k}\right)^{b_k - b_1}$ 인데, $a_1 \le a_k$이므로 이는 1보다 작다. $\therefore S' \ge S$이므로, 가정에 모순이 된다. 이를 모든 element에 대해 recursive하게 적용하면, S'이 optimal함을 증명할 수 있다.

2. ~~Red-Blue-coloring을 이용한다. cycle을 형성하는 red edge를 포함하지 않는 것중 max weight를 가지는 edge에 red, cut에서 blue edge를 포함하지 않는 min weight를 blue로 칠하는 것이다~~

T에 대해, T에서 edge $(u,v) \in E$를 제거함으로써 얻은 tree 두개를 T', T''이라 하자. 이 때의 G', G''에 대해 T'이 G'의 MST라 가정하자. 이때 $w(T) = w(u,v) + w(T') + w(T'')$ 인데, G'에서의 MST를 T_1'이라 하면 $\ge w(u,v) + w(T_1') + w(T'') = w(T_2)$. 이므로, T가 MST라는 가정에 모순이다. 따라서 T', T''은 각각 G', G''의 MST이고, $T' \subseteq T$, $G' \subseteq G$이다.

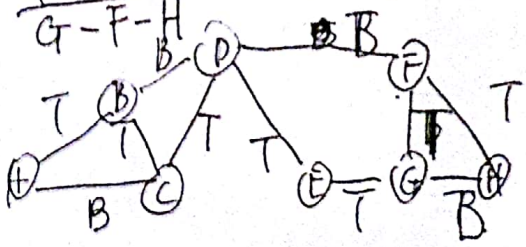
3. (a) 모든 node $u \in V$에 대해, $v \in V$와 u가 reachable하면 하나의 table에 넣을 수 있는 것이다. 따라서 BFS나 DFS를 이용한다. $u \in V$인 모든 node u에 대해 adjacency list를 갖고 있으면, BFS를 통해 visit한다. Visit한 node는 mark하고, 이렇게 하나의 node에서 reachable한 모든 node를 mark하여 하나의 Graph를 구성하는 set을 알 수 있다. 이후 unvisit인, mark되지 않은 모든 node에 대해 반복한 뒤 unique한 G의 개수를 세면 table의 개수를 알 수 있다. 모든 node와 edge가 한번씩 check되므로, run time은 $\Theta(V+E)$이다.

(b). Graph에 대해 홀수 개의 edge로 이뤄진 cycle이 존재하면 achive할 수 없다. table이 두 개이므로, 적어도 하나의 node는 다른 node와 edge가 있기 때문이다. 모든 node에 대해 iterating하면서 BFS를 시행한다. node가 visit될때 white로 mark하고, 만약 parent나 child가 white이면 black으로 mark한다. 만약 node가 visit된 상태에서 mark하고자 하는 rule과 current rule이 다르면 conflict가 발생한 것이므로 FALSE를 return한다. 모든 node에서 conflict가 없이 종료되면 TRUE를 return한다.

4. (a). BFS에서 adjacency list가 ~~order~~되어 있으므로, 다음 과정이 된다. A - B - D - F - H, A에서 B, D를 거친 이후, (E, F), (G, H)가 Depth가 같은 pair이므로 F의 child인 D가 E - G - H보다 먼저 detect된다.

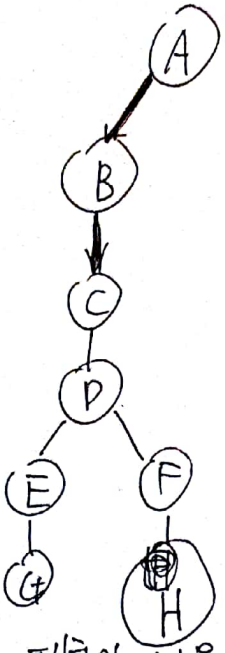
(b). ~~Recursive하게 adjacency list의 first element를 search한다. 따라서, A - B - C - D - E - F - G - H 가 된다.~~

Recursive하게 adjacency list에서 first element를 search하면, A - B - C - D - E - ~~F - G - H~~. 따라서 이에 해당하는 edge는 T, 이외는 모두 연결되어 B이다.



(b).

4. (차) (c). directed graph에서 아음과 같이 tree가 형성된다.



따라서 각각에 F, B, T, C를 표시하면 아음과 같다.