Algorithms: Homework 2

Yilin Zheng
jerryzyl18@gmail.com

Junfeng Liu
11510801@mail.sustc.edu.cn

August 4, 2017

Applying graph algorithm

Problem (a)

Find the shortest path way between s and t in the given graph.

Solution

Apply Dijkstra algorithm and we find:

- Path 1: $s \to b \to c \to t$
- Path 2: $s \to b \to d \to c \to t$
- Path 3: $s \to b \to d \to t$

All of the paths cost 105. ■

Problem (b)

Describe a graph that represents the scheduling problem and show that colouring that graph correctly.

Solution

Graph that represents the scheduling problem:

Since every faculty member may be part of more than one committees, so we can use **a vertex to represent a committee**. To avoid the committees with common members being scheduled to the same hour, we can use **an edge to connect any two vertices if the committees they represents have at least one common members**.

Scheduling with colours: Colour the vertices in the graph we generate with **as least as possible number of colours** but must keep **vertices that are adjacent with an edge** being coloured **in different colours**.

Then the numbers of colours we use is the hours wo spend to scheduling the meeting.

Let's take an **example** to demonstrate this method:

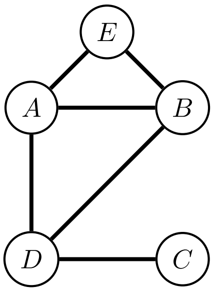
Assume committees: A, B, C, D, E

Members: $1, 2, 3, 4, 5, 6$

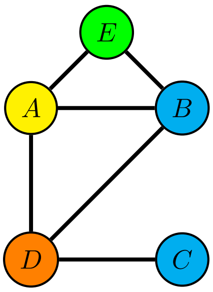
Members of committees:

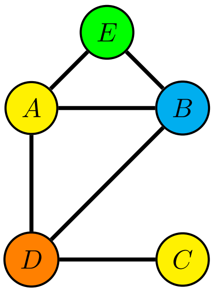
$$\begin{aligned} \{1,2\} \in A \quad & \{2,3\} \in B \\ \{4,5\} \in C \quad & \{2,4\} \in D \\ \{1,3,6\} \in E \quad & \end{aligned}$$

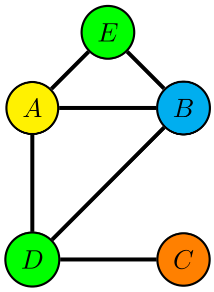
Then the graph we generate:



Use the colouring we get the coloured graphs:







Any of the three coloured graphs is the correct solution. The number of colour is 4, so there need at least four hours for these committees in this example. ■

Some simple graph properties

Problem (a)

Proof $2e/v \leq M$ if true otherwise provide a counterexample.

Solution

The statement is **true**.

Proof.

Assume d $(m \leq d \leq M)$ is the degree of vertex $v \in V$.
According to lemma:

$$\sum_i^v d_i = 2e \quad i = 1, 1, 3, 4, .., v$$

we have

$$2e \leq v \cdot M$$

which is

$$2e/v \leq M$$

■

Problem (b)

Proof $2e/v \geq m$ if true otherwise provide a counterexample.

Solution

The statement is also **true**.

Proof.

Like the step in previous solution, we still have

$$\sum_i^v d_i = 2e \quad i = 1, 1, 3, 4, .., v$$

then,

$$2e \geq v \cdot m$$

which is

$$2e/v \geq m$$

■

Problem (c)

Proof there exists a simple path (includes no cycles) of length at least m.

Proof.

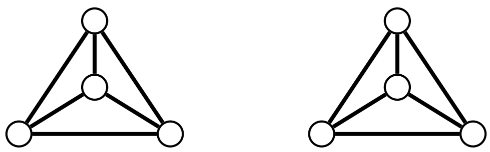
Pick any vertex, marked $v_0 \in V$ with degree k_0 $(m \leq k_0 \leq M)$ to start the path.
Then there are k_0 vertices adjacent to v_0, so choose any vertex v_1 of the k_0 vertices to extend the path, if it has $k_1(m \leq k_1 \leq M)$ degree, then we have $k_1 - 1$ choices to continue extending the length. Continue choosing a vertex v_2 from the $k_1 - 1$ vertices with degree $k_2(m \leq k_2 \leq M)$, then we still have at least $k_2 - 2$ vertices to select.
We can continue this argument until we've chosen v_m adjacent to v_{m-1} which is different from we've chosen $v_0, v_1, ..., v_{m-2}$.
Consider **the worse case** that for vertex v_m with degree $k_m(m \leq k_m \leq M)$, there leaves $k_m - m$ vertices to choose, which might be 0, then the path of length m comes to an end. But for **other cases**, there might still leave non-visited vertices so the length of the path can increase over m.
So we proven that the length of path is at least m. ■

Problem (d)

$m > 2$ implies the G is connected.

Solution

It is **false**.
A simple counterexample is that suppose there are two separated wheel graphs W and they both contain 4 vertices (W_4).
It is obvious that both of them have 3 edges and the minimum degree is $3(> 2)$, but this is **NOT** a connected graph.



■

Problem (e)

$M \leq v - 1$ if G is a simple graph.

Proof.

It is a **true**.
By definition, the simple graph means a graph with **no loop back** and **parallel edges**. So, each node can only connect **one** edge to at most $v - 1$.
In a simple graph with n nodes, the maximum degree must **less than or equal to** $n - 1$. ■

Some more graph properties

Problem (a)

Derive an expression in terms of i, j and m.

Solution

To generate the expression, we start from solving first subproblem.
To get the shortest distance between v_1 and v_j, we simulate **BFS algorithm** on the graph.
We firstly traverse vertices from v_1 to find its adjacent vertices v_2, v_{m+1}, v_{2m} and add to a search queue. Then sequentially pick up vertices from the queue and add their adjacent vertices into the queue (e.g., $v_3, v_{m+2}, v_m, v_{2m-1}$).
Repeat these steps until traversing to v_{2m}.
By this method we can easily generate the expression.
According to mathematics derivation, the distance between v_1 and v_j is:

$$\begin{cases} j-1, & 1 \le j < \frac{m}{2} + \frac{3}{2} \\ m+2-j, & \frac{m}{2} + \frac{3}{2} \le j < m+1 \\ j-m+\frac{1}{2}, & m+1 \le j < \frac{3}{2}m + \frac{1}{2} \\ 2m+1-j, & \frac{3}{2}m + \frac{1}{2} \le j \le 2m \end{cases}$$

■

Problem (b)

Prove that G^* is not 4-edge-connected.

Proof.

Since edges exist iff $i - j = 1 mod 2m \vee i - j = m mod 2m \vee i - j = 2m - 1 mod 2m$. So some vertices might have less than 4 adjacent vertices.
Take v_m for an example, it only has 3 adjacent vertices: v_{m-1}, v_{m+1}, v_{2m}.
So G^* is not a 4-edge-connected graph. ■

Path is a graph

Problem

Show that in every(simple) graph there is a path from any vertex of odd degree to some other vertex of odd degree.

Proof.

Assume a path starting with any vertex of odd degree ends with a vertex of even degree. Since the degree is even, then there must leave at least an edge which haven't been taken so that we can take to extent the path. So, the path must end with a vertex of odd degree. ■

Directed acyclic graphs

Problem (a)

Show that in any DAG, there is a vertex whose in-degree equals 0.

Proof.

Suppose there is no vertex with 0 in-degree, then for all vertices, their incoming edges should come from certain vertices which according to assumption should have both in-degree and out-degree. Then for this case, there must exist a directed cycle otherwise such graph does **NOT** exist. Since the graph we consider is **DAG** that cannot have any cycle, so the assumption is **false** and we prove that in any DAG there exist a vertex with 0 in-degree. ■

Problem (b)

Show that in any DAG, there is a vertex whose in-degree equals 0.

Proof.

Like the proof in the previous problem. Assume that there is no vertex with 0 out-degree, then we will get contradiction that there exists direct cycles while the graph should be a **DAG**. So there must exist a vertex with 0 out-degree. ■

Problem (c)

Show that in any DAG, one can order the vertices so as to respect edge directions. i.e., show that there exists a one-to-one and onto mapping $f : V(G) \to \{1, ..., n\}$ such that for every directed edge $(u, v), f(u) \le f(v)$. So every edge points from a lower numbered vertex to a higher numbered vertex.

Proof.

We prove this statement by induction. Since there exist a vertex called sink in the graph. If remove the sink v and leave a graph G' as induction hypothesis, in G' there exist one-to-one and onto mapping f'. Then for induction step, we add the sink v back and make $f(v) = f'(v)$ which v is **NOT** any vertices in $\{1, 2, 3..., n-1\}$ and for $x \in \{1, 2, 3, ..., n\}$ set $f(x) = n$. The statement still holds in this graph. ■

Problem (d)

Show that $V = L_0 \cup L_1 \cup \cdots \cup L_{k^*-1}$ is a stratification.

Proof.

For any vertex $v \in L_i$, $u \in L_j$ with $j \ge i$, so $u \in G_i$. Since L_i is the set of sources in G_i, then there are **no edges coming into the vertices in** G_i. Also, there has **no edge coming from** u **to** v. So, there exists no edge goes from a higher level to a lower level which means it is a stratification. ■

Problem (e)

Show that $k*$ in the previous part is the smallest $k \in \mathbb{N}$ such that there exists a stratification with k levels.

Proof.

To prove this statement, we can prove that for any $i \ge 0$, $\hat{G_i}$ is a subgraph of G_i.

We prove this equivalent proposition by induction.

Firstly assume any subgraph of G, marked as $\hat{G_i}$ for any $i \in \mathbb{N}$, with a stratification for the subgraph $\hat{L_0} \cup \hat{L_1} \cup \cdots \cup \hat{L_{k'-1}}$. Define $\hat{G_0} = G = G_0$ and $\hat{G_i} = \hat{G_{i-1}} \backslash \hat{L_{i-1}}$ for $i = 1, 2, 3, ...$.

- **Base case**: For $i = 0$, $\hat{G_0} = G_0$ since this is our definition, which
- **Induction hypothesis**: Suppose for any $i \le k (k \ge 0)$, the statement still holds.
- **Induction step**: For $i = k + 1$, since $\hat{L_k}$ is subset of the sources in $\hat{G_m}$, and L_m is the set of all sources in G_m, G_m is a subgraph of $\hat{G_m}$ then $G_m \backslash L_m$ is a subgraph of $\hat{G_m} \backslash \hat{L_m}$.

So, k^* above is the smallest $k \in \mathbb{N}$. ■

BFS–Proof of Correctness

Problem (a)

Show that, after BFS terminates, T is a connected subgraph of T.

Proof.

According to the Algorithm 1, we use mathematics induction to prove the equivalent statement that after BFS terminates, **vertices and edges in T exist in graph G** and the subgraph is connected.

- **Basic step**: For the starting point as **root**, it is obviously in graph G. The equivalent statement is **true**.
- **Induction hypothesis**: Suppose using Algorithm 1, after nth loop($n > 0$), all $k(k \in \mathbb{N} and k > 0)$ vertices and their adjacent edges in T exist in graph G.
- **Induction step**: In $(n+1)$th loop, we add vertices which are adjacent to the vertices added before and edges between them to the T. So, T still contains vertices and edges which also exist in G, and T is connected graph. The proposition is proven.

■

Problem (b)

Show that T spans G.

Proof.

If T spans G, then T contains all the vertices which exist in G, which means all the vertices in G have been visited after BFS terminates. So, we can prove all the vertices has been visited after BFS.
By contradiction, assume there exist one vertex which has NOT been visited. Then, there are two reasons:
(1) The unvisited vertex does **NOT** connect to any other vertices.
(2) The vertices connected to the unvisited vertex also have **NOT** be visited.
Consider these two cases. The reason(1) is impossible because G is a connected graph.
For reason(2), since its a connected graph, then BFS can continue and this is contradicted to the basic condition that we get T after BFS terminates. So there is no reason that there exist any unvisited vertex. Then the statement is proven. ■

Problem (c)

Show that T is a tree.

Proof.

We have proved that T is a connected subgraph of G. And by Algorithm 1, we will only visit unmarked neighbours and mark them. It means we can **NOT** visit a vertex **twice**. So there is no loop back in T. Because there is no loop back and its a connected subgraph of G, then T is a tree. ■

Problem (d)

Show that for every $y \in V(T)$, the path from x to y in T is a shortest path from x to y in G.

Proof.

By contradiction, assume there exists a vertex v. The path from root w to v is **NOT** the shortest path, then there exist a shorter path. We define p to be the real parent vertex of v, and p^* to be another parent vertex of v in the shorter path. According to Algorithm 1, we connected edge from v to p instead of v to p^* because p is ahead of p^* in the queue Q. There is a truth that distance of certain vertices to root increase with respect to its order in the queue.

$$
\begin{aligned}
distance(p) &< distance(p*) \\
path\_distance(root--p--x) &= distance(p) + c(c \geq 0) \\
path\_distance(root--p^*--x) &= distance(p*) + c \\
path\_distance(root--p--x) &< path\_distance(root--p^*--x)
\end{aligned}
$$

It contract with the assumption. So the path from x to y is a shortest path from x to y in graph G. ■