cs5800 algorithms s'22 — abhi shelat
due fri apr 15, 2022 at 11.59p via gradescope

H8.patel.nisargs

problem 1 *Number of shortest paths*

Given a graph $G = (V, E)$ with unit edge weights, and a starting node s, let $\delta(s, v)$ be the length of the shortest path between s and v (i.e., the smallest number of edges between s and v).

Define a new variable n_v which records the number of distinct shortest paths from s to v that have length $\delta(s, v)$ for every node in V. Design an algorithm that computes n_v for all nodes in V. Analyze the running time of the algorithm, and provide one-two sentence explanation for why the algorithm works.

Solution: We can solve this problem by modifying Breadth-First Search algorithm as follows:

CountShortestPaths(V, E, s)

```
for each u ∈ V − {s}
    (d_u, n_u) ← (∞, 0)
(d_s, n_s) ← (0, 1)
Q ← φ
Enqueue(Q, s)
while Q ≠ φ
    u ← Dequeue(Q)
    for each v ∈ Adj(u)
        if d_v > d_u + 1
            (d_v, n_v) ← (d_u + 1, n_u)
            Enqueue(Q, v)
        else if d_v = d_u + 1
            n_v ← n_v + n_u
return n
```

Time Complexity Analysis:
The running time complexity remains same as of BFS.
Time Complexity of CountShortestPaths(V, E, s) is $\Theta(V + E)$.

Correctness:
Let p be any shortest path from s to v then there exists a node u adjecent to v which is before v in $p(s \to \ldots \to u \to v)$ such that $\delta(s, v) = \delta(s, u) + 1$. Thus adding distinct shortest paths of each u satisfying the above condition, gives the distinct shortest paths for v.

PROBLEM 2 *Sparse graphs and short paths*

Let $G = (V, E)$ be a directed graph with edge weights $w(e)$ and no negative cycles.

1. State the run time of the All-pairs shortest path algorithm discussed in class.

2. Consider the following algorithm.

   AnotherShortest(G, w)

   1 Add a new node s' to G. Add edges of weight 0 from s' to every vertex $v \in V$. Call this new graph G'.
   2 Run BellmanFord(G', s') to produce shortest path lengths $\delta(s', v)$. If shortest paths are not well-defined, than halt.
   3 For each $e = (x, y) \in E$, set $w'(e) \leftarrow w(e) + \delta(s', x) - \delta(s', y)$
   4 For each $u \in V$, run Dijkstra(G, u, w') to compute $\delta(u, v)$ for all $v \in V$.
   5 Set $d_{u,v} \leftarrow \delta(u, v) - \delta(s', u) + \delta(s', v)$

   This problem will analyze what this algorithm does and why it works. The first step is to argue that the new edge weights w' that are defined in step (3) are all non-negative.

   Prove that for all $e \in E$, $w'(e) \geq 0$.

3. This explains why we can use the fast Dijkstra algorithm with edge weight w' in step (4) to compute shortest paths from node $v \in V$ to all other nodes in the graph. However, we must argue that the shortest paths under w' and under w will be the same shortest path.

   Prove that for any pairs of nodes $u, v \in V$, if p is a shortest path from u to v with respect to edge weight function w', then p is also a shortest path from u to v with respect to edge weight function w.

4. What is the running time of AnotherShortest in terms of V and E? When does this algorithm run faster than the All-pairs algorithm discussed in class?

Solution:

1. Floyd-Warshall All-pairs shortest path has time complexity $\Theta(V^3)$.

2. For any edge $e(x, y)$, applying triangle inequality, we have:

$$\begin{aligned} \delta(s', y) &\leq \delta(s', x) + w(e) \\ \delta(s', x) + w(e) - \delta(s', y) &\geq 0 \\ w'(e) &\geq 0 \end{aligned} \tag{1}$$

   Thus for all $e \in E$, $w'(e) \geq 0$.

3. For any path $p = (u, x_1, x_2, ..., x_n, v)$ computed on w', we have:

$$\begin{aligned}
w'(p) = \delta(u,v) &= w'(u,x_1) + w'(u,x_2) + ... + w'(x_n,v) \\
&= w(u,x_1) + \delta(s',u) - \delta(s',x_1) \\
&\quad + w(x_1,x_2) + \delta(s',x_1) - \delta(s',x_2) \\
&\quad + \dots \\
&\quad + w(x_n,v) + \delta(s',x_n) - \delta(s',v) \\
w'(p) &= w(p) + \delta(s',u) - \delta(s',v)
\end{aligned} \tag{2}$$

Let p be the shortest path between u and v calculated on w'. Let $p*$ be the shortest path between u and v computed on w. Then we have,

$$w(p*) \leq w(p) \tag{3}$$

And,

$$\begin{aligned}
w'(p) &\leq w'(p*) \\
w(p) + \delta(s',u) - \delta(s',v) &\leq w(p*) + \delta(s',u) - \delta(s',v) \\
w(p) &\leq w(p*)
\end{aligned} \tag{4}$$

Thus, from (3) and (4), we have $w(p) = w(p*)$. So p is also the shortest path between u and v computed on w. Hence proved.

4. **Time Complexity Analysis of** ANOTHERSHORTEST(G, w)**:**

   (a) Line 1 takes $\Theta(V)$. The new graph G' has $V' = V + 1$ vertices and $E' = E + V$ edges.

   (b) Line 2 runs BELLMANFORD(G', s') in $\Theta(V(E + V))$.

   (c) Line 3 runs in $\Theta(E)$.

   (d) Line 4 runs DIJKSTRA(G, u, w') for V nodes, runtime: $\Theta(V(V + E)\log(V))$.

   (e) Line 5 runs in $\Theta(V^2)$.

   The overall time complexity is predominated by line 4, $\Theta(V(V + E)\log(V))$.

   Comparision with Floyd–Warshall:
   This algorithm will have same time complexity as Floyd-Warshall when E is $\Theta(V^2/\log(V))$. Thus it will be faster for $E = \Theta(1)$ or $\Theta(V)$. Hence this algorithm works better on sparse graphs.

PROBLEM 3 *Edmonds-Karp shortest paths*

In class, we stated that in the Edmonds-Karp maxflow algorithm, the length of shortest paths in G are monotonically increasing. However, this is not obvious because as we add augmenting paths, new edges are introduced to the graph. In this problem, we will prove the following:

Lemma 1 *For any $j > i$ and for any $u \in V$, $\delta_j(s, u) \geq \delta_i(s, u)$.*

The proof will be by contradiction. Suppose not, for the sake of contradiction. Let i be the first time that the shortest path distance to some node decreases after pushing flow along the i^{th} augmentation. Moreover, let v be the *node with the smallest* distance to s at $i + 1$ for which $\delta_i(s, v) > \delta_{i+1}(s, v)$. Let p_i, p_{i+1} be respective shortest paths from s to v at times i and $i + 1$.

Each answer should be roughly 1 sentence. You may refer to steps (1)–(7) in your explanations.

1. Define node u to be the node that occurs before v on path p_{i+1}. The first claim is that $\delta_{i+1}(s, u) \geq \delta_i(s, u)$. Why does this follow? (one sentence)

2. Next, explain why $\delta_{i+1}(s, v) = \delta_{i+1}(s, u) + 1$.

3. Explain why edge $e_{i+1} = (u, v)$ did not exist in the graph at time i.

4. Thus, the edge e_{i+1} must have been added after the i flow, which implies that the augmenting path at i took the form $s \rightsquigarrow v \rightarrow u \rightsquigarrow t$, i.e., that pushed flow from v to u. Explain why at time i, we have

$$\delta_i(s, u) = \delta_i(s, v) + 1$$

5. Explain why (4) implies

$$\delta_{i+1}(s, u) \geq \delta_i(s, v) + 1$$

6. Adding one to each side, we have

$$\delta_{i+1}(s, u) + 1 \geq \delta_i(s, v) + 2$$

   Explain why this implies that

$$\delta_{i+1}(s, v) \geq \delta_i(s, v) + 2$$

7. Explain why the last statement in (6) is a contradiction.

Solution:

1. Since at $i+1$, v is the node with smallest distance to s for which $\delta_{i+1}(s,v) < \delta_i(s,v)$, and u occurs before v on path p_{i+1}, then

$$\delta_{i+1}(s,u) \geq \delta_i(s,u)$$

2. As u is the node that occurs just before v on shortest path p_{i+1},

$$\delta_{i+1}(s,v) = \delta_{i+1}(s,u) + 1$$

3. Now from the given condition, we have:

$$\begin{aligned} &\delta_i(s,v) > \delta_{i+1}(s,v) \\ &\delta_i(s,v) > \delta_{i+1}(s,u) + 1 \quad \text{From 2} \\ &\delta_i(s,v) > \delta_i(s,u) + 1 \quad \text{From 1} \end{aligned} \tag{5}$$

   If edge $e_{i+1} = (u,v)$ exist at time i, then $\delta_i(s,v) \leq \delta_i(s,u) + 1$ based on triangle inequality which contradicts the above result. Hence edge $e_{i+1} = (u,v)$ does not exist at time i.

4. Based on result of 3, edge $e_{i+1} = (u,v)$ must be added after the i flow, and the augmenting path at i took $s \rightsquigarrow v \rightarrow u \rightsquigarrow t$.Thus v comes before u at time i. Hence,

$$\delta_i(s,u) = \delta_i(s,v) + 1$$

5. Since, $\delta_{i+1}(s,u) \geq \delta_i(s,u)$, (From 1). Substituting 5 in it, we have:

$$\delta_{i+1}(s,u) \geq \delta_i(s,v) + 1$$

6.

$$\begin{aligned} \delta_{i+1}(s,u) &\geq \delta_i(s,v) + 1 \quad \text{From 5} \\ \delta_{i+1}(s,u) + 1 &\geq \delta_i(s,v) + 2 \\ \delta_{i+1}(s,v) &\geq \delta_i(s,v) + 2 \quad \text{From 2} \end{aligned} \tag{6}$$

7. The last equation in 6 contradicts our initial assumption: $\delta_{i+1}(s,v) < \delta_i(s,v)$.
   Thus for every node v, we have $\delta_{i+1}(s,v) \geq \delta_i(s,v)$ for any time i.
   Let $j > i$, then we have $\delta_j(s,v) \geq \delta_{j-1}(s,v) \geq ... \geq \delta_{i+1}(s,v) \geq \delta_i(s,v)$.
   Hence our lemma holds.