CMPSCI 611: Advanced Algorithms

Lecture 10: Seidel's Algorithm

Andrew McGregor

Last Compiled: October 8, 2009

Outline

Seidel's Algorithm

Seidel's Algorithm

Problem: For an undirected, unweighted graph G, compute $\delta_G(i, j)$ for all $i, j \in V$.

Seidel's Algorithm

Problem: For an undirected, unweighted graph G, compute $\delta_G(i,j)$ for all $i,j \in V$.

Seidel's Algorithm is based on matrix multiplication and runs in time

$$O(\mu(n) \log n)$$

where $\mu(n)$ is the time to multiply two $n \times n$ matrices together. Recall

$$n^2 \leq \mu(n) \leq n^{2.38}$$

Seidel's Algorithm

Problem: For an undirected, unweighted graph G, compute $\delta_G(i,j)$ for all $i,j \in V$.

Seidel's Algorithm is based on matrix multiplication and runs in time

$$O(\mu(n) \log n)$$

where $\mu(n)$ is the time to multiply two $n \times n$ matrices together. Recall

$$n^2 \leq \mu(n) \leq n^{2.38}$$

Definition

Let M_G be the *adjacency matrix* of $G = (V, E)$, i.e., an $n \times n$ binary matrix where

$$M_G(i,j) = 1 \text{ iff } (i,j) \in E$$

The G_2 graph

Definition

Given a undirected, unweighted graph $G = (V, E)$, define $G_2 = (V, E')$ where $(i, j) \in E'$ iff $\delta_G(i, j) \leq 2$.

The G_2 graph

Definition

Given a undirected, unweighted graph $G = (V, E)$, define $G_2 = (V, E')$ where $(i, j) \in E'$ iff $\delta_G(i, j) \leq 2$.

Lemma

Let $P_G(i, j) = 1$ *if* $\delta_G(i, j)$ *is odd and* $P_G(i, j) = 0$ *otherwise. Then,*

$$\delta_G(i, j) = 2\delta_{G_2}(i, j) - P_G(i, j)$$

The G_2 graph

Definition

Given a undirected, unweighted graph $G = (V, E)$, define $G_2 = (V, E')$ where $(i, j) \in E'$ iff $\delta_G(i, j) \leq 2$.

Lemma

Let $P_G(i, j) = 1$ if $\delta_G(i, j)$ is odd and $P_G(i, j) = 0$ otherwise. Then,

$$\delta_G(i, j) = 2\delta_{G_2}(i, j) - P_G(i, j)$$

Corollary

If $M_{G_2}(i, j) = 1$ for all $i \neq j$, then

$$\delta_G(i, j) = \begin{cases} 0 & \text{if } i = j \\ 1 & \text{if } M_G(i, j) = 1 \\ 2 & \text{otherwise} \end{cases}$$

Seidel's Algorithm

Algorithm (Seidel(M_G))

1. *compute* M_{G_2}
2. *if* $\forall i \neq j : M_{G_2}(i,j) = 1$, *return*

$$D_G[i,j] = \begin{cases} 0 & \textit{if } i = j \\ 1 & \textit{if } M_G(i,j) = 1 \\ 2 & \textit{otherwise} \end{cases}$$

3. *else:*
   - 3.1 *compute* $D'_G = \text{Seidel}(M_{G_2})$
   - 3.2 *compute* P_G
   - 3.3 *return* $D_G = 2D'_G - P_G$

Seidel's Algorithm

Algorithm (Seidel(M_G))

1. *compute* M_{G_2}
2. *if* $\forall i \neq j : M_{G_2}(i,j) = 1$, *return*

$$D_G[i,j] = \begin{cases} 0 & \textit{if } i = j \\ 1 & \textit{if } M_G(i,j) = 1 \\ 2 & \textit{otherwise} \end{cases}$$

3. *else:*
   3.1 *compute* $D'_G = \text{Seidel}(M_{G_2})$
   3.2 *compute* P_G
   3.3 *return* $D_G = 2D'_G - P_G$

Mystery Steps: How can we compute M_{G_2} and P_G quickly?

Proof of Correctness

Definition

The diameter of a graph G is the "longest shortest path",

$$\operatorname{diam}(G) = \max_{i,j} \delta_G(i, j)$$

Proof of Correctness

Definition

The diameter of a graph G is the "longest shortest path",

$$\text{diam}(G) = \max_{i,j} \delta_G(i,j)$$

1. Proof by induction on the diameter of graph.

Proof of Correctness

Definition

The diameter of a graph G is the "longest shortest path",

$$\operatorname{diam}(G) = \max_{i,j} \delta_G(i,j)$$

1. Proof by induction on the diameter of graph.
2. Base case: Earlier corollary proved result when diameter is at most 2

Proof of Correctness

Definition

The diameter of a graph G is the "longest shortest path",

$$\text{diam}(G) = \max_{i,j} \delta_G(i,j)$$

1. Proof by induction on the diameter of graph.
2. Base case: Earlier corollary proved result when diameter is at most 2
3. Note that if $\text{diam}(G) \geq 3$:

$$\text{diam}(G_2) \leq \text{diam}(G)/2 + 1/2 < \text{diam}(G)$$

Proof of Correctness

Definition

The diameter of a graph G is the "longest shortest path",

$$\text{diam}(G) = \max_{i,j} \delta_G(i,j)$$

1. Proof by induction on the diameter of graph.
2. Base case: Earlier corollary proved result when diameter is at most 2
3. Note that if $\text{diam}(G) \geq 3$:

$$\text{diam}(G_2) \leq \text{diam}(G)/2 + 1/2 < \text{diam}(G)$$

4. Induction hypothesis: $D'_G = \text{Seidel}(M_{G_2})$ is correctly computed.

Proof of Correctness

Definition

The diameter of a graph G is the "longest shortest path",

$$\text{diam}(G) = \max_{i,j} \delta_G(i, j)$$

1. Proof by induction on the diameter of graph.
2. Base case: Earlier corollary proved result when diameter is at most 2
3. Note that if $\text{diam}(G) \geq 3$:

$$\text{diam}(G_2) \leq \text{diam}(G)/2 + 1/2 < \text{diam}(G)$$

4. Induction hypothesis: $D'_G = \text{Seidel}(M_{G_2})$ is correctly computed.
5. By earlier lemma, D_G is computed correctly.

Computing M_{G_2}

Lemma

$$M_{G_2}(i,j) = \begin{cases} 1 & \textit{if } i \neq j \textit{ and } (M_G(i,j) = 1 \textit{ or } M_G^2(i,j) > 0) \\ 0 & \textit{otherwise} \end{cases}$$

Computing M_{G_2}

Lemma

$$M_{G_2}(i,j) = \begin{cases} 1 & \textit{if } i \neq j \textit{ and } (M_G(i,j) = 1 \textit{ or } M_G^2(i,j) > 0) \\ 0 & \textit{otherwise} \end{cases}$$

Proof.

$M_G^2(i,j) = \sum_k M_G(i,k)M_G(k,j) = \#$ of length 2 paths from i to j. □

Computing M_{G_2}

Lemma

$$M_{G_2}(i,j) = \begin{cases} 1 & \textit{if } i \neq j \textit{ and } (M_G(i,j) = 1 \textit{ or } M_G^2(i,j) > 0) \\ 0 & \textit{otherwise} \end{cases}$$

Proof.

$M_G^2(i,j) = \sum_k M_G(i,k) M_G(k,j) = \#$ of length 2 paths from i to j. □

Can compute M_{G_2} in $O(\mu(n))$ time.

Computing P_G

P_G can be computed in $O(\mu(n))$ time...

Computing P_G

P_G can be computed in $O(\mu(n))$ time...

Lemma

Let $X = D'_G M_G$ *where* $D'_G(i,j) = \delta_{G_2}(i,j)$. *Then,*

$$P_G(i,j) = 0 \iff X(i,j) \geq \delta_{G_2}(i,j)\,\mathrm{degree}_G(j)$$

where $\mathrm{degree}_G(j)$ *is the number of edges incident to node* j *in graph* G.

Computing P_G

P_G can be computed in $O(\mu(n))$ time...

Lemma

Let $X = D'_G M_G$ *where* $D'_G(i,j) = \delta_{G_2}(i,j)$*. Then,*

$$P_G(i,j) = 0 \iff X(i,j) \geq \delta_{G_2}(i,j)\,\mathrm{degree}_G(j)$$

where $\mathrm{degree}_G(j)$ *is the number of edges incident to node* j *in graph* G*.*

Note that: $X(i,j) = \sum_k \delta_{G_2}(i,k) M_G(k,j) = \sum_{k \in \mathrm{Adj}_G(j)} \delta_{G_2}(i,k)$

Computing P_G

P_G can be computed in $O(\mu(n))$ time...

Lemma

Let $X = D'_G M_G$ *where* $D'_G(i,j) = \delta_{G_2}(i,j)$. *Then,*

$$P_G(i,j) = 0 \Longleftrightarrow X(i,j) \geq \delta_{G_2}(i,j)\,\mathrm{degree}_G(j)$$

where $\mathrm{degree}_G(j)$ *is the number of edges incident to node* j *in graph* G*.*

Note that: $X(i,j) = \sum_k \delta_{G_2}(i,k) M_G(k,j) = \sum_{k \in \mathrm{Adj}_G(j)} \delta_{G_2}(i,k)$

It's sufficient to prove:

$$(P_G(i,j) = 0) \Rightarrow\ (\forall k \in \mathrm{Adj}_G(j) : \delta_{G_2}(i,k) \geq \delta_{G_2}(i,j))$$

$$\begin{aligned}(P_G(i,j) = 1) \Rightarrow\ & (\forall k \in \mathrm{Adj}_G(j) : \delta_{G_2}(i,k) \leq \delta_{G_2}(i,j)) \\ & \text{and } (\exists k \in \mathrm{Adj}_G(j) : \delta_{G_2}(i,k) < \delta_{G_2}(i,j))\end{aligned}$$

First Part

- Assume $P_G(i, j) = 0$. Need to show that for $k \in \mathrm{Adj}_G(j)$,

$$\delta_{G_2}(i, k) \geq \delta_{G_2}(i, j)$$

First Part

- Assume $P_G(i,j) = 0$. Need to show that for $k \in \mathrm{Adj}_G(j)$,

$$\delta_{G_2}(i,k) \geq \delta_{G_2}(i,j)$$

- If k is not on shortest path from i to j:

$$\delta_G(i,k) \geq \delta_G(i,j)$$

First Part

- Assume $P_G(i,j) = 0$. Need to show that for $k \in \mathrm{Adj}_G(j)$,

$$\delta_{G_2}(i,k) \geq \delta_{G_2}(i,j)$$

- If k is not on shortest path from i to j:

$$2\delta_{G_2}(i,k) \geq \delta_G(i,k) \geq \delta_G(i,j)$$

First Part

- Assume $P_G(i,j) = 0$. Need to show that for $k \in \mathrm{Adj}_G(j)$,

$$\delta_{G_2}(i,k) \geq \delta_{G_2}(i,j)$$

- If k is not on shortest path from i to j:

$$2\delta_{G_2}(i,k) \geq \delta_G(i,k) \geq \delta_G(i,j) = 2\delta_{G_2}(i,j)$$

First Part

- Assume $P_G(i,j) = 0$. Need to show that for $k \in \text{Adj}_G(j)$,

$$\delta_{G_2}(i,k) \geq \delta_{G_2}(i,j)$$

- If k is not on shortest path from i to j:

$$2\delta_{G_2}(i,k) \geq \delta_G(i,k) \geq \delta_G(i,j) = 2\delta_{G_2}(i,j)$$

- If k is on shortest path from i to j:

$$\delta_G(i,j) = \delta_G(i,k) + 1$$

First Part

- Assume $P_G(i, j) = 0$. Need to show that for $k \in \mathrm{Adj}_G(j)$,

$$\delta_{G_2}(i, k) \geq \delta_{G_2}(i, j)$$

- If k is not on shortest path from i to j:

$$2\delta_{G_2}(i, k) \geq \delta_G(i, k) \geq \delta_G(i, j) = 2\delta_{G_2}(i, j)$$

- If k is on shortest path from i to j:

$$2\delta_{G_2}(i, j) = \delta_G(i, j) = \delta_G(i, k) + 1$$

First Part

- Assume $P_G(i,j) = 0$. Need to show that for $k \in \mathrm{Adj}_G(j)$,

$$\delta_{G_2}(i,k) \geq \delta_{G_2}(i,j)$$

- If k is not on shortest path from i to j:

$$2\delta_{G_2}(i,k) \geq \delta_G(i,k) \geq \delta_G(i,j) = 2\delta_{G_2}(i,j)$$

- If k is on shortest path from i to j:

$$2\delta_{G_2}(i,j) = \delta_G(i,j) = \delta_G(i,k) + 1 = 2\delta_{G_2}(i,k) - 1 + 1 = 2\delta_{G_2}(i,k)$$

Second Part

- Assume $P_G(i,j) = 1$. Need to show that for $k \in \mathrm{Adj}_G(j)$,

$$\delta_{G_2}(i,k) \leq \delta_{G_2}(i,j)$$

  and for at least one k, the inequality is strict.

Second Part

- Assume $P_G(i, j) = 1$. Need to show that for $k \in \mathrm{Adj}_G(j)$,

  $$\delta_{G_2}(i, k) \leq \delta_{G_2}(i, j)$$

  and for at least one k, the inequality is strict.
- If k is on shortest path from i to j:

  $$\delta_G(i, k) = \delta_G(i, j) - 1$$

Second Part

- Assume $P_G(i,j) = 1$. Need to show that for $k \in \mathrm{Adj}_G(j)$,

$$\delta_{G_2}(i,k) \leq \delta_{G_2}(i,j)$$

  and for at least one k, the inequality is strict.
- If k is on shortest path from i to j:

$$2\delta_{G_2}(i,k) = \delta_G(i,k) = \delta_G(i,j) - 1$$

Second Part

- Assume $P_G(i,j) = 1$. Need to show that for $k \in \mathrm{Adj}_G(j)$,

$$\delta_{G_2}(i,k) \leq \delta_{G_2}(i,j)$$

  and for at least one k, the inequality is strict.
- If k is on shortest path from i to j:

$$2\delta_{G_2}(i,k) = \delta_G(i,k) = \delta_G(i,j) - 1 = 2\delta_{G_2}(i,j) - 2$$

Second Part

- Assume $P_G(i,j) = 1$. Need to show that for $k \in \mathrm{Adj}_G(j)$,

$$\delta_{G_2}(i,k) \leq \delta_{G_2}(i,j)$$

  and for at least one k, the inequality is strict.
- If k is on shortest path from i to j:

$$2\delta_{G_2}(i,k) = \delta_G(i,k) = \delta_G(i,j) - 1 = 2\delta_{G_2}(i,j) - 2$$

- If k is not on shortest path from i to j:

$$\delta_{G_2}(i,j) = \delta_{G_2}(i,j)/2 + 1/2$$

$$\delta_{G_2}(i,k) \leq \delta_{G_2}(i,k)/2{+}1/2 \leq (\delta_{G_2}(i,j){+}1)/2{+}1/2 = \delta_{G_2}(i,j)/2{+}1$$

Total Running Time

Algorithm (Seidel(M_G))

1. *compute* M_{G_2}
2. *if* $\forall i \neq j : M_{G_2}(i,j) = 1$, *return*

$$D_G[i,j] = \begin{cases} 0 & \textit{if } i \neq j \\ 1 & \textit{if } M_G(i,j) = 1 \\ 2 & \textit{if otherwise} \end{cases}$$

3. *else:*
   - 3.1 *compute* $D'_G = \text{Seidel}(M_{G_2})$
   - 3.2 *compute* P_G
   - 3.3 *return* $D_G = 2D'_G - P_G$

Total Running Time

Algorithm (Seidel(M_G))

1. *compute* M_{G_2}
2. *if* $\forall i \neq j : M_{G_2}(i,j) = 1$, *return*

$$D_G[i,j] = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } M_G(i,j) = 1 \\ 2 & \text{if otherwise} \end{cases}$$

3. *else:*
   - 3.1 *compute* $D'_G = \text{Seidel}(M_{G_2})$
   - 3.2 *compute* P_G
   - 3.3 *return* $D_G = 2D'_G - P_G$

Running Time: $O(\mu(n) \log n)$ since depth of recursion is $O(\log n)$ and each iteration takes $O(\mu(n))$.

For Next Time...

- Finish reading up Section 5.3 of the notes.
- No class on Tuesday!
- Homework 2 due a week from today (minus 75 minutes).