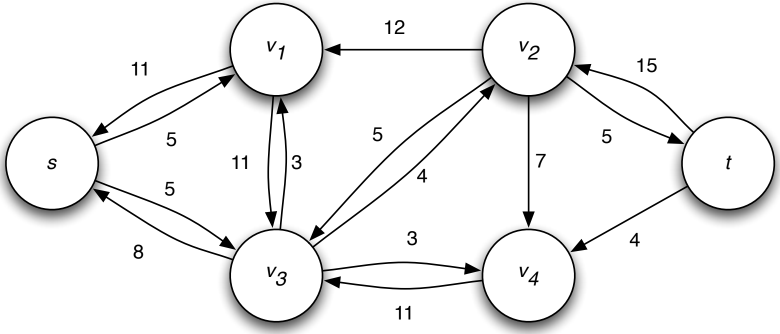
CMPSCI 611: Advanced Algorithms

Lecture 13: Finishing Network Flow. Randomized Algorithms

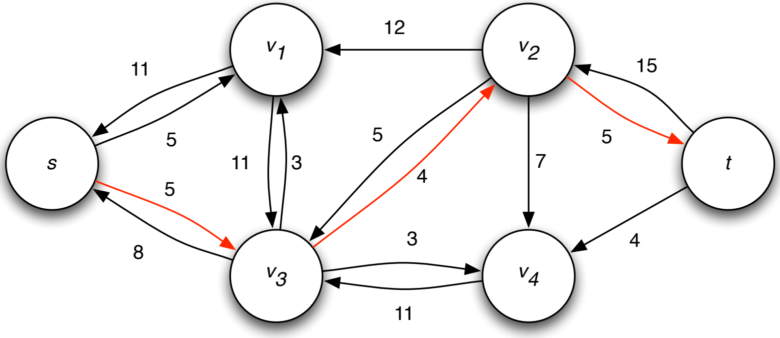
Andrew McGregor

Last Compiled: April 4, 2024

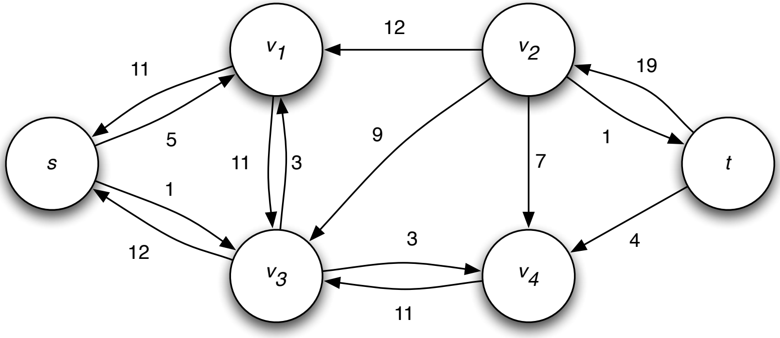
Residual



Augmenting Path



New Residual Graph



Ford-Fulkerson Algorithm with Edmonds-Karp Heuristic

Algorithm

1. *flow* $f = 0$
2. *while there exists an augmenting path* p *for* f
   2.1 *find shortest (unweighted) augmenting path* p
   2.2 *augment* f *by* $b(p)$ *units along* p
3. *return* f

Theorem

The algorithms finds a maximum flow in time $O(|E|^2|V|)$

Proof of Running Time (1/3)

Definition

Let $\delta_f(s, u)$ be length of shortest unweighted path from s to u in the G_f.

Definition

(u, v) is critical if it's on augmenting path p for f and $C_f(u, v) = b(p)$.

Lemma

$\delta_f(s, v)$ is non-decreasing as f changes.

Lemma

Between occasions when (u, v) is critical, $\delta_f(s, u)$ increases by at least 2.

Proof of Running Time.

- Max distance in G_f is $|V|$ so any edge is critical at most $1 + |V|/2$ times
- At most $2|E|$ edges in residual network
- There's a critical edge in each iteration so $O(|E||V|)$ iterations
- Each iteration takes $O(|E|)$ to find shortest path

□

Proof of Running Time (2/3)

Lemma

$\delta_f(s, v)$ is non-decreasing as f changes.

Proof.

- Consider augmenting f to f'
- For contradiction, pick v that minimizes $\delta_{f'}(s, v)$ subject to:

$$\delta_{f'}(s, v) < \delta_f(s, v)$$

- Let u be vertex before v on shortest s-v path in $G_{f'}$. Note $\delta_{f'}(s, u) \geq \delta_f(s, u)$ and $\delta_{f'}(s, v) = \delta_{f'}(s, u) + 1$
- Claim $(u, v) \notin E_f$
  - Otherwise $\delta_f(s, v) \leq \delta_f(s, u) + 1$
  - $\delta_f(s, u) \leq \delta_{f'}(s, u)$ implies $\delta_f(s, v) \leq \delta_{f'}(s, u) + 1 = \delta_{f'}(s, v)$.
  - Contradicts $\delta_{f'}(s, v) < \delta_f(s, v)$.
- $(u, v) \notin E_f$ and $(u, v) \in E_{f'}$ implies augmentation contains (v, u)
- Since augmentation was shortest path:

$$\delta_f(s, v) = \delta_f(s, u) - 1 \leq \delta_{f'}(s, u) - 1 = \delta_{f'}(s, v) - 2$$

Proof of Running Time (3/3)

Lemma

Between occasions when (u, v) *is critical,* $\delta_f(s, u)$ *increases by at least 2.*

Proof.

- Let (u, v) be critical in the augmentation of f
- Since (u, v) on shortest path: $\delta_f(s, u) = \delta_f(s, v) - 1$
- After augmentation (u, v) disappears from residual network!
- Let f'' be next flow with $(u, v) \in G_{f''}$ and f' be flow right before f''
- $(u, v) \notin G_{f'}$ but $(u, v) \in G_{f''}$ implies (v, u) used to augment f'
- Therefore $\delta_{f'}(s, v) = \delta_{f'}(s, u) - 1$ and so

$$\delta_f(s, u) = \delta_f(s, v) - 1 \leq \delta_{f'}(s, v) - 1 = \delta_{f'}(s, u) - 2$$

□

Probability Refresher

- Expectation of random variable:

$$\mathbb{E}[X] = \sum_r r\mathbb{P}[X = r]$$

- Linearity of expectation:

$$\mathbb{E}[X + Y] = \mathbb{E}[X] + \mathbb{E}[Y]$$

- Conditional Probability: For arbitrary events A and B,

$$\mathbb{P}[A|B] = \mathbb{P}[A \cap B] / \mathbb{P}[B]$$

and $\mathbb{P}[\cap_{i=1}^n A_i] = \mathbb{P}[A_1]\mathbb{P}[A_2|A_1] \dots \mathbb{P}\left[A_n| \cap_{i=1}^{n-1} A_i\right]$

Quicksort

Problem: Sort an array of distinct values $X = [x_1, \ldots, x_n]$

Algorithm

1. *Pick a pivot $x \in X$ at random from the array*
2. *Construct new arrays $Y = [y_1, \ldots, y_k]$, $Z = [z_1, \ldots, z_{n-k-1}]$ where*

$$y < x < z \text{ for all } y \in Y, z \in Z$$

3. *Recursively sort Y and Z to get Y' and Z'*
4. *Return the array that concatenates Y', x, and Z'*

What's the expected number of comparisons performed in this algorithm?

Probability two items are compared

Lemma

Let a and b be the i-th and j-th smallest element of X where $i < j$.

$$\Pr[a \textit{ is compared to } b] = \frac{2}{j - i + 1}$$

Proof.

1. Consider $S = \{x \in X : a \leq x \leq b\}$
2. a and b are compared iff the first pivot chosen from S is either a or b
3. Elements of S are equally likely to be chosen as a pivot, so

$$\Pr[a \text{ is compared to } b] = \frac{2}{|S|} = \frac{2}{j - i + 1}$$

□

Expected Number of Comparisons

Lemma

Expected number of comparisons performs is $O(n \log n)$.

Proof.

1. Let $Z_{ij} = 1$ if the i-th smallest element is compared to j-th smallest element and $Z_{ij} = 0$ otherwise.
2. Number of comparisons: $\sum_{1 \leq i < j \leq n} Z_{ij}$
3. Expected number of comparisons:

$$\mathbb{E}\left[\sum_{1 \leq i < j \leq n} Z_{ij}\right] = \sum_{1 \leq i < j \leq n} \mathbb{E}\left[Z_{ij}\right] = \sum_{1 \leq i < j \leq n} \frac{2}{j - i + 1} = \sum_{j=2}^{n} \sum_{k=2}^{j} \frac{2}{k}$$

4. Because $H_n = 1 + 1/2 + 1/3 + \ldots + 1/n = O(\log n)$,

$$\mathbb{E}\left[\sum_{1 \leq i < j \leq n} Z_{ij}\right] \leq \sum_{j=2}^{n} \sum_{k=2}^{n} \frac{2}{k} = n \cdot O(\log n) = O(n \log n)$$

□