Implementation of algorithms and data structures

9. seminar

Jirka Fink

https://ktiml.mff.cuni.cz/~fink/

Department of Theoretical Computer Science and Mathematical Logic
Faculty of Mathematics and Physics
Charles University in Prague

Winter semestr 2025/26

Last change 26. listopadu 2025

Maximum flow in a network

Network (graph)

- (V, E) is a directed graph on n vertices and m edges
- $c : E \to \mathbb{R}_0^+$ is capacity of edges
- $s, t \in V$ are source and sink vertices

Network (graph)

- (V, E) is a directed graph on n vertices and m edges
- $c : E \to \mathbb{R}_0^+$ is capacity of edges
- $s, t \in V$ are source and sink vertices

Flow

Flow is a function $f : E \to \mathbb{R}_0^+$ satisfying:

- Capacity constraint: $0 \leq f(e) \leq c(e)$ for every edge e
- Kirchhoff law: $\sum_{u:uv \in E} f(uv) = \sum_{u:vu \in E} f(vu)$ for every vertex v except s, t

Maximum flow in a network

Network (graph)

- (V, E) is a directed graph on n vertices and m edges
- $c : E \to \mathbb{R}_0^+$ is capacity of edges
- $s, t \in V$ are source and sink vertices

Flow

Flow is a function $f : E \to \mathbb{R}_0^+$ satisfying:

- Capacity constraint: $0 \leq f(e) \leq c(e)$ for every edge e
- Kirchhoff law: $\sum_{u:uv \in E} f(uv) = \sum_{u:vu \in E} f(vu)$ for every vertex v except s, t

Overflow

Overflow of a vertex v is $f^{\triangle}(v) = \sum_{u:uv \in E} f(uv) - \sum_{u:vu \in E} f(vu)$

Maximum flow in a network

Network (graph)

- (V, E) is a directed graph on n vertices and m edges
- $c : E \to \mathbb{R}_0^+$ is capacity of edges
- $s, t \in V$ are source and sink vertices

Flow

Flow is a function $f : E \to \mathbb{R}_0^+$ satisfying:

- Capacity constraint: $0 \leq f(e) \leq c(e)$ for every edge e
- Kirchhoff law: $\sum_{u:uv \in E} f(uv) = \sum_{u:vu \in E} f(vu)$ for every vertex v except s, t

Overflow

Overflow of a vertex v is $f^{\triangle}(v) = \sum_{u:uv \in E} f(uv) - \sum_{u:vu \in E} f(vu)$

The problem of maximum flow in a network

For a given network, find a flow maximizing overflow of the sink $f^{\triangle}(s)$.

Terminology

- Reserve (residual) of an edge uv is $r(uv) = c(uv) - f(uv) + f(vu)$
- Edge uv is saturated if $r(uv) = 0$
- Path is saturated if at least one edge of the path is saturated

Terminology

- Reserve (residual) of an edge uv is $r(uv) = c(uv) - f(uv) + f(vu)$
- Edge uv is saturated if $r(uv) = 0$
- Path is saturated if at least one edge of the path is saturated

Remark

We assume that for every edge $uv \in E$ there exists a reverse edge $vu \in E$ since we can add the edge vu into E with zero capacity $c(vu) = 0$.

Ford-Fulkerson

Terminology

- Reserve (residual) of an edge uv is $r(uv) = c(uv) - f(uv) + f(vu)$
- Edge uv is saturated if $r(uv) = 0$
- Path is saturated if at least one edge of the path is saturated

Remark

We assume that for every edge $uv \in E$ there exists a reverse edge $vu \in E$ since we can add the edge vu into E with zero capacity $c(vu) = 0$.

Theorem

Flow f is maximal if and only if all paths from the source to the sink are saturated.

Terminology

- Reserve (residual) of an edge uv is $r(uv) = c(uv) - f(uv) + f(vu)$
- Edge uv is saturated if $r(uv) = 0$
- Path is saturated if at least one edge of the path is saturated

Remark

We assume that for every edge $uv \in E$ there exists a reverse edge $vu \in E$ since we can add the edge vu into E with zero capacity $c(vu) = 0$.

Theorem

Flow f is maximal if and only if all paths from the source to the sink are saturated.

Ford-Fulkerson algorithm

Start with zero flow f. While there exists a non-saturated path P from the source to the sink, increase the flow f on edges of P by $\min\{r(e);\ e \in P\}$.

Optimality

Cut

- Consider $A \subset V$ containing the source but not the sink
- Cut $E(A) = \{uv \in E;\ u \in A,\ v \notin A\}$ is the set of edges from A to $V \setminus A$
- Cut A is saturated if all edges of $E(A)$ are saturated

Optimality

Cut

- Consider $A \subset V$ containing the source but not the sink
- Cut $E(A) = \{uv \in E;\ u \in A,\ v \notin A\}$ is the set of edges from A to $V \setminus A$
- Cut A is saturated if all edges of $E(A)$ are saturated

Theorem

Flow f is maximal if and only if there exists a saturated cut.

Optimality

Cut

- Consider $A \subset V$ containing the source but not the sink
- Cut $E(A) = \{uv \in E;\ u \in A,\ v \notin A\}$ is the set of edges from A to $V \setminus A$
- Cut A is saturated if all edges of $E(A)$ are saturated

Theorem

Flow f is maximal if and only if there exists a saturated cut.

Finding saturated cut

Start with $A = \{s\}$. While there exists a non-saturated edge uv with $u \in A$ and $v \notin A$, insert v into A.

Optimality

Cut

- Consider $A \subset V$ containing the source but not the sink
- Cut $E(A) = \{uv \in E;\ u \in A,\ v \notin A\}$ is the set of edges from A to $V \setminus A$
- Cut A is saturated if all edges of $E(A)$ are saturated

Theorem

Flow f is maximal if and only if there exists a saturated cut.

Finding saturated cut

Start with $A = \{s\}$. While there exists a non-saturated edge uv with $u \in A$ and $v \notin A$, insert v into A.

Testing correctness of a solution

- Capacity constraint
- Kirchhoff law
- Saturated cut

Wave

Wave is a function $f : E \to \mathbb{R}_0^+$ satisfying:

- Capacity constraint: $0 \leq f(e) \leq c(e)$ for every edge e
- Non-negative overflows: $f^\triangle(v) \geq 0$ for every vertex v except the source

Wave

Wave is a function $f : E \to \mathbb{R}_0^+$ satisfying:

- Capacity constraint: $0 \leq f(e) \leq c(e)$ for every edge e
- Non-negative overflows: $f^\triangle(v) \geq 0$ for every vertex v except the source

Operation: Overflow transfer

Transferring overflow on an edge uv means increasing $f(uv)$ by $\min\left\{f^\triangle(u), r(uv)\right\}$.

Goldberg's algorithm

Wave

Wave is a function $f : E \to \mathbb{R}_0^+$ satisfying:

- Capacity constraint: $0 \le f(e) \le c(e)$ for every edge e
- Non-negative overflows: $f^\triangle(v) \ge 0$ for every vertex v except the source

Operation: Overflow transfer

Transferring overflow on an edge uv means increasing $f(uv)$ by $\min\left\{f^\triangle(u), r(uv)\right\}$.

Height

- Height is a function $h : V \to \mathbb{Z}_0^+$

Goldberg's algorithm

Wave

Wave is a function $f : E \to \mathbb{R}_0^+$ satisfying:

- Capacity constraint: $0 \leq f(e) \leq c(e)$ for every edge e
- Non-negative overflows: $f^\triangle(v) \geq 0$ for every vertex v except the source

Operation: Overflow transfer

Transferring overflow on an edge uv means increasing $f(uv)$ by $\min\left\{f^\triangle(u), r(uv)\right\}$.

Height

- Height is a function $h : V \to \mathbb{Z}_0^+$
- We are allowed to transfer overflow only from a higher vertex to a lower one (downhill)

Goldberg's algorithm

Wave

Wave is a function $f : E \rightarrow \mathbb{R}_0^+$ satisfying:

- Capacity constraint: $0 \leq f(e) \leq c(e)$ for every edge e
- Non-negative overflows: $f^{\triangle}(v) \geq 0$ for every vertex v except the source

Operation: Overflow transfer

Transferring overflow on an edge uv means increasing $f(uv)$ by $\min\left\{f^{\triangle}(u), r(uv)\right\}$.

Height

- Height is a function $h : V \rightarrow \mathbb{Z}_0^+$
- We are allowed to transfer overflow only from a higher vertex to a lower one (downhill)
- Invariant: $h(s) = n$ a $h(t) = 0$

Wave

Wave is a function $f : E \to \mathbb{R}_0^+$ satisfying:

- Capacity constraint: $0 \leq f(e) \leq c(e)$ for every edge e
- Non-negative overflows: $f^\triangle(v) \geq 0$ for every vertex v except the source

Operation: Overflow transfer

Transferring overflow on an edge uv means increasing $f(uv)$ by $\min\left\{f^\triangle(u), r(uv)\right\}$.

Height

- Height is a function $h : V \to \mathbb{Z}_0^+$
- We are allowed to transfer overflow only from a higher vertex to a lower one (downhill)
- Invariant: $h(s) = n$ a $h(t) = 0$
- Height of other vertices is initialized by 0 and algorithm can only increase it (by 1)

Wave

Wave is a function $f : E \to \mathbb{R}_0^+$ satisfying:

- Capacity constraint: $0 \leq f(e) \leq c(e)$ for every edge e
- Non-negative overflows: $f^\triangle(v) \geq 0$ for every vertex v except the source

Operation: Overflow transfer

Transferring overflow on an edge uv means increasing $f(uv)$ by $\min\left\{f^\triangle(u), r(uv)\right\}$.

Height

- Height is a function $h : V \to \mathbb{Z}_0^+$
- We are allowed to transfer overflow only from a higher vertex to a lower one (downhill)
- Invariant: $h(s) = n$ a $h(t) = 0$
- Height of other vertices is initialized by 0 and algorithm can only increase it (by 1)
- If for a vertex v with $f^\triangle(v) > 0$ no overflow can be transferred, increase height $h(v)$ by one

Goldberg's algorithm

$$h(v) = \begin{cases} n & \text{for } v = s \\ 0 & \text{otherwise} \end{cases}$$

$$f(uv) = \begin{cases} c(uv) & \text{for } u = s \\ 0 & \text{otherwise} \end{cases}$$

while *exists a vertex* $u \neq s, t$ *satisfying* $f^{\triangle}(u) > 0$ **do**

  if *exists an edge* uv *satisfying* $r(uv) > 0$ *and* $h(u) > h(v)$ **then**

    transfer overflow on edge uv

  else

    increase height $h(u)$ by 1

$$h(v) = \begin{cases} n & \text{for } v = s \\ 0 & \text{otherwise} \end{cases}$$

$$f(uv) = \begin{cases} c(uv) & \text{for } u = s \\ 0 & \text{otherwise} \end{cases}$$

while *exists a vertex* $u \neq s, t$ *satisfying* $f^{\triangle}(u) > 0$ **do**
    if *exists an edge* uv *satisfying* $r(uv) > 0$ *and* $h(u) > h(v)$ **then**
        transfer overflow on edge uv
    else
        increase height $h(u)$ by 1

Invariants satisfied in every iteration

- Function f is a wave

Analysis of Goldberg's algorithm

```
h(v) = { n   for v = s
       { 0   otherwise
f(uv) = { c(uv)   for u = s
        { 0       otherwise

while exists a vertex u ≠ s,t satisfying f^△(u) > 0 do
    if exists an edge uv satisfying r(uv) > 0 and h(u) > h(v) then
        transfer overflow on edge uv
    else
        increase height h(u) by 1
```

Invariants satisfied in every iteration

- Function f is a wave
- $h(s) = n$ a $h(t) = 0$

Analysis of Goldberg's algorithm

$h(v) = \begin{cases} n & \text{for } v = s \\ 0 & \text{otherwise} \end{cases}$

$f(uv) = \begin{cases} c(uv) & \text{for } u = s \\ 0 & \text{otherwise} \end{cases}$

while *exists a vertex* $u \neq s, t$ *satisfying* $f^{\triangle}(u) > 0$ **do**
    if *exists an edge* uv *satisfying* $r(uv) > 0$ *and* $h(u) > h(v)$ **then**
        transfer overflow on edge uv
    else
        increase height $h(u)$ by 1

Invariants satisfied in every iteration

- Function f is a wave
- $h(s) = n$ a $h(t) = 0$
- Height of vertices never decreases

Analysis of Goldberg's algorithm

$h(v) = \begin{cases} n & \text{for } v = s \\ 0 & \text{otherwise} \end{cases}$

$f(uv) = \begin{cases} c(uv) & \text{for } u = s \\ 0 & \text{otherwise} \end{cases}$

while *exists a vertex* $u \neq s, t$ *satisfying* $f^{\triangle}(u) > 0$ **do**
  if *exists an edge* uv *satisfying* $r(uv) > 0$ *and* $h(u) > h(v)$ **then**
    transfer overflow on edge uv
  else
    increase height $h(u)$ by 1

Invariants satisfied in every iteration

- Function f is a wave
- $h(s) = n$ a $h(t) = 0$
- Height of vertices never decreases
- For every edge uv if $r(uv) > 0$ then $h(u) \leq h(v) + 1$

Invariant: Paths to the sink

From every vertex u with $f^{\triangle}(u) > 0$ there exists a non-saturated path to the source.

Invariant: Paths to the sink

From every vertex u with $f^{\triangle}(u) > 0$ there exists a non-saturated path to the source.

- Let $A = \{v$: there exists a non-saturated path from u to $v\}$

Invariant: Paths to the sink

From every vertex u with $f^{\triangle}(u) > 0$ there exists a non-saturated path to the source.

- Let $A = \{v : \text{there exists a non-saturated path from } u \text{ to } v\}$

$$\sum_{v \in A} f^{\triangle}(v) = \sum_{ba \in E, a \in A, b \notin A} f(ba) - \sum_{ab \in E, a \in A, b \notin A} f(ab) \leq 0$$

since $r(ab) = f(ba) = 0$ for edges $ba \in E$ where $a \in A$ and $b \notin A$

Invariant: Paths to the sink

From every vertex u with $f^{\triangle}(u) > 0$ there exists a non-saturated path to the source.

- Let $A = \{v : \text{there exists a non-saturated path from } u \text{ to } v\}$

$$\sum_{v \in A} f^{\triangle}(v) = \sum_{ba \in E, a \in A, b \notin A} f(ba) - \sum_{ab \in E, a \in A, b \notin A} f(ab) \leq 0$$

  since $r(ab) = f(ba) = 0$ for edges $ba \in E$ where $a \in A$ and $b \notin A$
- A contains u with $f^{\triangle}(u) > 0$ and the source is the only vertex with negative overflow

Analysis of Goldberg's algorithm

Invariant: Paths to the sink

From every vertex u with $f^{\triangle}(u) > 0$ there exists a non-saturated path to the source.

- Let $A = \{v : \text{there exists a non-saturated path from } u \text{ to } v\}$

$$\sum_{v \in A} f^{\triangle}(v) = \sum_{ba \in E, a \in A, b \notin A} f(ba) - \sum_{ab \in E, a \in A, b \notin A} f(ab) \leq 0$$

  since $r(ab) = f(ba) = 0$ for edges $ba \in E$ where $a \in A$ and $b \notin A$
- A contains u with $f^{\triangle}(u) > 0$ and the source is the only vertex with negative overflow
- A contains the source

Analysis of Goldberg's algorithm

Invariant: Paths to the sink

From every vertex u with $f^\triangle(u) > 0$ there exists a non-saturated path to the source.

- Let $A = \{v : \text{there exists a non-saturated path from } u \text{ to } v\}$

$$\sum_{v \in A} f^\triangle(v) = \sum_{ba \in E, a \in A, b \notin A} f(ba) - \sum_{ab \in E, a \in A, b \notin A} f(ab) \le 0$$

  since $r(ab) = f(ba) = 0$ for edges $ba \in E$ where $a \in A$ and $b \notin A$
- A contains u with $f^\triangle(u) > 0$ and the source is the only vertex with negative overflow
- A contains the source

Invariant: For every vertex u holds $h(u) \le 2n$

- When $h(u)$ is increased above $2n$, then $f^\triangle(u) > 0$, so there exists non-saturated path from u to the source which contains an edge with gradient at least 2 which is a contradition.

Analysis of Goldberg's algorithm

Invariant: Paths to the sink

From every vertex u with $f^{\triangle}(u) > 0$ there exists a non-saturated path to the source.

- Let $A = \{v : \text{there exists a non-saturated path from } u \text{ to } v\}$

$$\sum_{v \in A} f^{\triangle}(v) = \sum_{ba \in E, a \in A, b \notin A} f(ba) - \sum_{ab \in E, a \in A, b \notin A} f(ab) \leq 0$$

  since $r(ab) = f(ba) = 0$ for edges $ba \in E$ where $a \in A$ and $b \notin A$
- A contains u with $f^{\triangle}(u) > 0$ and the source is the only vertex with negative overflow
- A contains the source

Invariant: For every vertex u holds $h(u) \leq 2n$

- When $h(u)$ is increased above $2n$, then $f^{\triangle}(u) > 0$, so there exists non-saturated path from u to the source which contains an edge with gradient at least 2 which is a contradition.
- Corollary: Height is increased at most $2n^2$-times

The number of overflow transfers

- Overflow transfer is called saturated if the reserve is reduced to zero

The number of overflow transfers

- Overflow transfer is called saturated if the reserve is reduced to zero
- The number of saturated transfers is at most nm

The number of overflow transfers

- Overflow transfer is called saturated if the reserve is reduced to zero
- The number of saturated transfers is at most nm
- The number of non-saturated transfers is at most $O(n^2 m)$

Analysis of Goldberg's algorithm

The number of overflow transfers

- Overflow transfer is called saturated if the reserve is reduced to zero
- The number of saturated transfers is at most nm
- The number of non-saturated transfers is at most $O(n^2 m)$
  - Consider the potential $\sum_{u:f^\triangle(u)>0} h(u)$

The number of overflow transfers

- Overflow transfer is called saturated if the reserve is reduced to zero
- The number of saturated transfers is at most nm
- The number of non-saturated transfers is at most $O(n^2 m)$
  - Consider the potential $\sum_{u:f^\triangle(u)>0} h(u)$
  - Increasing the height increases the potential by one

The number of overflow transfers

- Overflow transfer is called saturated if the reserve is reduced to zero
- The number of saturated transfers is at most nm
- The number of non-saturated transfers is at most $O(n^2 m)$
  - Consider the potential $\sum_{u: f^{\triangle}(u) > 0} h(u)$
  - Increasing the height increases the potential by one
  - Saturated transfer increases the potential by at most $2n$

The number of overflow transfers

- Overflow transfer is called saturated if the reserve is reduced to zero
- The number of saturated transfers is at most nm
- The number of non-saturated transfers is at most $O(n^2 m)$
  - Consider the potential $\sum_{u: f^{\triangle}(u) > 0} h(u)$
  - Increasing the height increases the potential by one
  - Saturated transfer increases the potential by at most $2n$
  - In total, the potential is increased by $O(n^2 m)$, it starts from zero, and it is always non-negative

The number of overflow transfers

- Overflow transfer is called saturated if the reserve is reduced to zero
- The number of saturated transfers is at most nm
- The number of non-saturated transfers is at most $O(n^2 m)$
  - Consider the potential $\sum_{u: f^{\triangle}(u)>0} h(u)$
  - Increasing the height increases the potential by one
  - Saturated transfer increases the potential by at most $2n$
  - In total, the potential is increased by $O(n^2 m)$, it starts from zero, and it is always non-negative
  - Non-saturated transfer decreases the potential by at least 1

Analysis of Goldberg's algorithm

The number of overflow transfers

- Overflow transfer is called saturated if the reserve is reduced to zero
- The number of saturated transfers is at most nm
- The number of non-saturated transfers is at most $O(n^2 m)$
  - Consider the potential $\sum_{u: f^\triangle(u) > 0} h(u)$
  - Increasing the height increases the potential by one
  - Saturated transfer increases the potential by at most $2n$
  - In total, the potential is increased by $O(n^2 m)$, it starts from zero, and it is always non-negative
  - Non-saturated transfer decreases the potential by at least 1

Algorithm always terminate and it returns a maximum flow

- Algorithm terminates after $O(n^2 m)$ steps

Analysis of Goldberg's algorithm

The number of overflow transfers

- Overflow transfer is called saturated if the reserve is reduced to zero
- The number of saturated transfers is at most nm
- The number of non-saturated transfers is at most $O(n^2 m)$
  - Consider the potential $\sum_{u:f^{\triangle}(u)>0} h(u)$
  - Increasing the height increases the potential by one
  - Saturated transfer increases the potential by at most $2n$
  - In total, the potential is increased by $O(n^2 m)$, it starts from zero, and it is always non-negative
  - Non-saturated transfer decreases the potential by at least 1

Algorithm always terminate and it returns a maximum flow

- Algorithm terminates after $O(n^2 m)$ steps
- Algorithm returns a flow since it terminates if all vertices (except s, t) has zero overflow.

The number of overflow transfers

- Overflow transfer is called saturated if the reserve is reduced to zero
- The number of saturated transfers is at most nm
- The number of non-saturated transfers is at most $O(n^2 m)$
  - Consider the potential $\sum_{u:f^{\triangle}(u)>0} h(u)$
  - Increasing the height increases the potential by one
  - Saturated transfer increases the potential by at most $2n$
  - In total, the potential is increased by $O(n^2 m)$, it starts from zero, and it is always non-negative
  - Non-saturated transfer decreases the potential by at least 1

Algorithm always terminate and it returns a maximum flow

- Algorithm terminates after $O(n^2 m)$ steps
- Algorithm returns a flow since it terminates if all vertices (except s, t) has zero overflow.
- Resulting flow is maximum
  - Otherwise there exists a non-saturated st-path and one of its edge has gradient 2.

Faster version of Goldberd's algorithm

Improvement

- From all vertices with positive overflow, choose the highest one.

Faster version of Goldberd's algorithm

Improvement

- From all vertices with positive overflow, choose the highest one.
- Time complexity is reduced to $O(n^2\sqrt{m})$

Improvement

- From all vertices with positive overflow, choose the highest one.
- Time complexity is reduced to $O(n^2\sqrt{m})$
- The proof is in literature

Theoretical questions

- What happens if the network is not connected?

Exercises

Theoretical questions

- What happens if the network is not connected?
- What happens if the network is connected, but it is not strongly connected?

Theoretical questions

- What happens if the network is not connected?
- What happens if the network is connected, but it is not strongly connected?
- Let A be the set of all vertices having height at least n when the algorithm terminated. Does $E(A)$ forms a saturated cut between source and sink?

Exercises

Theoretical questions

- What happens if the network is not connected?
- What happens if the network is connected, but it is not strongly connected?
- Let A be the set of all vertices having height at least n when the algorithm terminated. Does $E(A)$ forms a saturated cut between source and sink?
- Is the source the only vertex of height n?

Exercises

Theoretical questions

- What happens if the network is not connected?
- What happens if the network is connected, but it is not strongly connected?
- Let A be the set of all vertices having height at least n when the algorithm terminated. Does $E(A)$ forms a saturated cut between source and sink?
- Is the source the only vertex of height n?
- What may happen if we set the height of source to be $n-1$ (or $n-2$)?

Theoretical questions

- What happens if the network is not connected?
- What happens if the network is connected, but it is not strongly connected?
- Let A be the set of all vertices having height at least n when the algorithm terminated. Does $E(A)$ forms a saturated cut between source and sink?
- Is the source the only vertex of height n?
- What may happen if we set the height of source to be $n - 1$ (or $n - 2$)?
- For which graphs the algorithm requires the largest number of iterations (for fixed n or m)?

Exercises

Theoretical questions

- What happens if the network is not connected?
- What happens if the network is connected, but it is not strongly connected?
- Let A be the set of all vertices having height at least n when the algorithm terminated. Does $E(A)$ forms a saturated cut between source and sink?
- Is the source the only vertex of height n?
- What may happen if we set the height of source to be $n-1$ (or $n-2$)?
- For which graphs the algorithm requires the largest number of iterations (for fixed n or m)?

Implementation questions

- Develop unit and fuzz tests

Exercises

Theoretical questions

- What happens if the network is not connected?
- What happens if the network is connected, but it is not strongly connected?
- Let A be the set of all vertices having height at least n when the algorithm terminated. Does $E(A)$ forms a saturated cut between source and sink?
- Is the source the only vertex of height n?
- What may happen if we set the height of source to be $n - 1$ (or $n - 2$)?
- For which graphs the algorithm requires the largest number of iterations (for fixed n or m)?

Implementation questions

- Develop unit and fuzz tests
- Based on our analysis, develop as many data consistency tests as possible

Exercises

Theoretical questions

- What happens if the network is not connected?
- What happens if the network is connected, but it is not strongly connected?
- Let A be the set of all vertices having height at least n when the algorithm terminated. Does $E(A)$ forms a saturated cut between source and sink?
- Is the source the only vertex of height n?
- What may happen if we set the height of source to be $n-1$ (or $n-2$)?
- For which graphs the algorithm requires the largest number of iterations (for fixed n or m)?

Implementation questions

- Develop unit and fuzz tests
- Based on our analysis, develop as many data consistency tests as possible
- Find data representation so that whole algorithm has complexity $O(n^2 m)$

Exercises

Theoretical questions

- What happens if the network is not connected?
- What happens if the network is connected, but it is not strongly connected?
- Let A be the set of all vertices having height at least n when the algorithm terminated. Does $E(A)$ forms a saturated cut between source and sink?
- Is the source the only vertex of height n?
- What may happen if we set the height of source to be $n - 1$ (or $n - 2$)?
- For which graphs the algorithm requires the largest number of iterations (for fixed n or m)?

Implementation questions

- Develop unit and fuzz tests
- Based on our analysis, develop as many data consistency tests as possible
- Find data representation so that whole algorithm has complexity $O(n^2 m)$
- How to find a highest vertex with positive overflow to improve the complexity to $O(n^2 \sqrt{m})$?