15.5
Algorithms via Basic Search
Algorithms via Basic Search - I

1. Given $G$ and nodes $u$ and $v$, can $u$ reach $v$?
2. Given $G$ and $u$, compute $rch(u)$.

Use $Explore(G, u)$ to compute $rch(u)$ in $O(n + m)$ time.
Algorithms via Basic Search - II

Given $G$ and $u$, compute all $v$ that can reach $u$, that is all $v$ such that $u \in rch(v)$. Naive: $O(n(n + m))$

Definition (Reverse graph.)

Given $G = (V, E)$, $G^{rev}$ is the graph with edge directions reversed $G^{rev} = (V, E')$ where $E' = \{(y, x) \mid (x, y) \in E\}$

Compute $rch(u)$ in $G^{rev}!$

- Correctness: exercise
- Running time: $O(n + m)$ to obtain $G^{rev}$ from $G$ and $O(n + m)$ time to compute $rch(u)$ via Basic Search. If both $Out(v)$ and $In(v)$ are available at each $v$ then no need to explicitly compute $G^{rev}$. Can do $Explore(G, u)$ in $G^{rev}$ implicitly.
Given $G$ and $u$, compute all $v$ that can reach $u$, that is all $v$ such that $u \in rch(v)$. Naive: $O(n(n + m))$

Definition (Reverse graph.)

Given $G = (V, E)$, $G^{rev}$ is the graph with edge directions reversed $G^{rev} = (V, E')$ where $E' = \{(y, x) \mid (x, y) \in E\}$

Compute $rch(u)$ in $G^{rev}$!

- **Correctness**: exercise

- **Running time**: $O(n + m)$ to obtain $G^{rev}$ from $G$ and $O(n + m)$ time to compute $rch(u)$ via Basic Search. If both $Out(v)$ and $In(v)$ are available at each $v$ then no need to explicitly compute $G^{rev}$. Can do $Explore(G, u)$ in $G^{rev}$ implicitly.
Given $G$ and $u$, compute all $v$ that can reach $u$, that is all $v$ such that $u \in \text{rch}(v)$.  
Naive: $O(n(n + m))$

**Definition (Reverse graph.)**

Given $G = (V, E)$, $G^{\text{rev}}$ is the graph with edge directions reversed $G^{\text{rev}} = (V, E')$ where $E' = \{(y, x) \mid (x, y) \in E\}$

Compute $\text{rch}(u)$ in $G^{\text{rev}}$!

**Correctness:** exercise

**Running time:** $O(n + m)$ to obtain $G^{\text{rev}}$ from $G$ and $O(n + m)$ time to compute $\text{rch}(u)$ via Basic Search. If both $\text{Out}(v)$ and $\text{In}(v)$ are available at each $v$ then no need to explicitly compute $G^{\text{rev}}$. Can do $\text{Explore}(G, u)$ in $G^{\text{rev}}$ implicitly.
Given $G$ and $u$, compute all $v$ that can reach $u$, that is all $v$ such that $u \in rch(v)$. Naive: $O(n(n + m))$

**Definition (Reverse graph.)**

Given $G = (V, E)$, $G^{rev}$ is the graph with edge directions reversed $G^{rev} = (V, E')$ where $E' = \{(y, x) | (x, y) \in E\}$

Compute $rch(u)$ in $G^{rev}$!

**Correctness:** exercise

**Running time:** $O(n + m)$ to obtain $G^{rev}$ from $G$ and $O(n + m)$ time to compute $rch(u)$ via Basic Search. If both $Out(v)$ and $In(v)$ are available at each $v$ then no need to explicitly compute $G^{rev}$. Can do $Explore(G, u)$ in $G^{rev}$ implicitly.
Algorithms via Basic Search - III

\( \text{SCC}(G, u) = \{v \mid u \text{ is strongly connected to } v\} \)

- Find the strongly connected component containing node \( u \). That is, compute \( \text{SCC}(G, u) \).

\[
\text{SCC}(G, u) = \text{rch}(G, u) \cap \text{rch}(G^{\text{rev}}, u)
\]

Hence, \( \text{SCC}(G, u) \) can be computed with \textit{Explore}(G, u) and \textit{Explore}(G^{\text{rev}}, u).

Total \( O(n + m) \) time.

Why can \( \text{rch}(G, u) \cap \text{rch}(G^{\text{rev}}, u) \) be done in \( O(n) \) time?
Find the strongly connected component containing node $u$. That is, compute $\text{SCC}(G, u)$.

$\text{SCC}(G, u) = \{v \mid u \text{ is strongly connected to } v\}$

Hence, $\text{SCC}(G, u)$ can be computed with $\text{Explore}(G, u)$ and $\text{Explore}(G^{rev}, u)$. Total $O(n + m)$ time.

Why can $\text{rch}(G, u) \cap \text{rch}(G^{rev}, u)$ be done in $O(n)$ time?
Find the strongly connected component containing node $u$. That is, compute $\text{SCC}(G, u)$.

$\text{SCC}(G, u) = \{v \mid u \text{ is strongly connected to } v\}$

$\text{SCC}(G, u) = \text{rch}(G, u) \cap \text{rch}(G^{\text{rev}}, u)$

Hence, $\text{SCC}(G, u)$ can be computed with $\text{Explore}(G, u)$ and $\text{Explore}(G^{\text{rev}}, u)$. Total $O(n + m)$ time.

Why can $\text{rch}(G, u) \cap \text{rch}(G^{\text{rev}}, u)$ be done in $O(n)$ time?
SCC\((G, u)\) = \{v | \text{u is strongly connected to } v\}

Find the strongly connected component containing node \(u\). That is, compute SCC\((G, u)\).

SCC\((G, u)\) = \text{rch}(G, u) \cap \text{rch}(G^{\text{rev}}, u)

Hence, SCC\((G, u)\) can be computed with Explore\((G, u)\) and Explore\((G^{\text{rev}}, u)\). Total \(O(n + m)\) time.

Why can \(\text{rch}(G, u) \cap \text{rch}(G^{\text{rev}}, u)\) be done in \(O(n)\) time?
SCC I: Graph $G$ and its reverse graph $G^{\text{rev}}$
SCC II: Graph $G$ a vertex $F$ .. and its reachable set $rch(G, F)$

Graph $G$

Reachable set of vertices from $F$
SCC III: Graph $G$ a vertex $F$ 

.. and the set of vertices that can reach it in $G$: $rch(G^{rev}, F)$

Graph $G$

Set of vertices that can reach $F$, computed via **DFS** in the reverse graph $G^{rev}$. 
SCC IV: Graph $G$ a vertex $F$ and...

its strong connected component in $G$: $\text{SCC}(G, F)$

$\text{SCC}(G, F) = \text{rch}(G, F) \cap \text{rch}(G^{\text{rev}}, F)$
Is $G$ strongly connected?

Pick arbitrary vertex $u$. Check if $\text{SCC}(G, u) = V$. 
Is \( G \) strongly connected?

Pick arbitrary vertex \( u \). Check if \( \text{SCC}(G, u) = V \).
Find all strongly connected components of $G$.

While $G$ is not empty do
Pick arbitrary node $u$
find $S = \text{SCC}(G, u)$
Remove $S$ from $G$

Question: Why doesn’t removing one strong connected components affect the other strong connected components?

Running time: $O(n(n + m))$.

Question: Can we do it in $O(n + m)$ time?
Find all strongly connected components of $G$.

While $G$ is not empty do
Pick arbitrary node $u$
find $S = SCC(G, u)$
Remove $S$ from $G$

Question: Why doesn’t removing one strongly connected components affect the other strongly connected components?

Running time: $O(n(n + m))$.

Question: Can we do it in $O(n + m)$ time?
Find all strongly connected components of $G$.

While $G$ is not empty do
  Pick arbitrary node $u$
  find $S = \text{SCC}(G, u)$
  Remove $S$ from $G$

Question: Why doesn’t removing one strong connected components affect the other strong connected components?

Running time: $O(n(n + m))$.

Question: Can we do it in $O(n + m)$ time?
Find all strongly connected components of $G$.

While $G$ is not empty do
   Pick arbitrary node $u$
   find $S = SCC(G, u)$
   Remove $S$ from $G$

**Question:** Why doesn’t removing one strong connected components affect the other strong connected components?

Running time: $O(n(n + m))$.

**Question:** Can we do it in $O(n + m)$ time?
Find all strongly connected components of \( G \).

While \( G \) is not empty do
  Pick arbitrary node \( u \)
  find \( S = \text{SCC}(G, u) \)
  Remove \( S \) from \( G \)

**Question:** Why doesn’t removing one strong connected components affect the other strong connected components?

Running time: \( O(n(n + m)) \).

**Question:** Can we do it in \( O(n + m) \) time?
THE END

...

(for now)