C — Customs Controls

Problem

Given a vertex-weighted graph, color k vertices red and $n - k$ vertices blue such that every shortest path from 1 to n has a monochromatic edge.

Solution

C — Customs Controls

Problem

Given a vertex-weighted graph, color k vertices red and $n-k$ vertices blue such that every shortest path from 1 to n has a monochromatic edge.

Solution

1. Dijkstra's algorithm finds all edges that are part of some shortest path from 1 to n.

C — Customs Controls

Problem

Given a vertex-weighted graph, color k vertices red and $n-k$ vertices blue such that every shortest path from 1 to n has a monochromatic edge.

Solution

1. Dijkstra's algorithm finds all edges that are part of some shortest path from 1 to n.
2. These edges form a directed acyclic graph. Find a topological ordering.

C — Customs Controls

Problem

Given a vertex-weighted graph, color k vertices red and $n-k$ vertices blue such that every shortest path from 1 to n has a monochromatic edge.

Solution

1. Dijkstra's algorithm finds all edges that are part of some shortest path from 1 to n.
2. These edges form a directed acyclic graph. Find a topological ordering.
3. Color the first k vertices in the ordering red, and the remaining ones blue:
   - A shortest path from 1 to n now only switches between red and blue once, so every shortest path on 3 or more vertices must have a monochromatic edge.

C — Customs Controls

Problem

Given a vertex-weighted graph, color k vertices red and $n-k$ vertices blue such that every shortest path from 1 to n has a monochromatic edge.

Solution

1. Dijkstra's algorithm finds all edges that are part of some shortest path from 1 to n.
2. These edges form a directed acyclic graph. Find a topological ordering.
3. Color the first k vertices in the ordering red, and the remaining ones blue:
   - A shortest path from 1 to n now only switches between red and blue once, so every shortest path on 3 or more vertices must have a monochromatic edge.
4. Special case: this does not work if there is a direct edge from 1 to n.

C — Customs Controls

Problem

Given a vertex-weighted graph, color k vertices red and $n - k$ vertices blue such that every shortest path from 1 to n has a monochromatic edge.

Solution

1. Dijkstra's algorithm finds all edges that are part of some shortest path from 1 to n.
2. These edges form a directed acyclic graph. Find a topological ordering.
3. Color the first k vertices in the ordering red, and the remaining ones blue:
   - A shortest path from 1 to n now only switches between red and blue once, so every shortest path on 3 or more vertices must have a monochromatic edge.
4. Special case: this does not work if there is a direct edge from 1 to n.
   - Since graph is vertex-weighted, edge from 1 to n is the *only* shortest path.
   - We only need to make sure 1 and n get the same color.

C — Customs Controls

Problem

Given a vertex-weighted graph, color k vertices red and $n - k$ vertices blue such that every shortest path from 1 to n has a monochromatic edge.

Solution

1. Dijkstra's algorithm finds all edges that are part of some shortest path from 1 to n.
2. These edges form a directed acyclic graph. Find a topological ordering.
3. Color the first k vertices in the ordering red, and the remaining ones blue:
   - A shortest path from 1 to n now only switches between red and blue once, so every shortest path on 3 or more vertices must have a monochromatic edge.
4. Special case: this does not work if there is a direct edge from 1 to n.
   - Since graph is vertex-weighted, edge from 1 to n is the *only* shortest path.
   - We only need to make sure 1 and n get the same color.
   - Always possible, except if $n = 2$ and $k = 1$.

C — Customs Controls

Problem

Given a vertex-weighted graph, color k vertices red and $n - k$ vertices blue such that every shortest path from 1 to n has a monochromatic edge.

Solution

1. Dijkstra's algorithm finds all edges that are part of some shortest path from 1 to n.
2. These edges form a directed acyclic graph. Find a topological ordering.
3. Color the first k vertices in the ordering red, and the remaining ones blue:
   - A shortest path from 1 to n now only switches between red and blue once, so every shortest path on 3 or more vertices must have a monochromatic edge.
4. Special case: this does not work if there is a direct edge from 1 to n.
   - Since graph is vertex-weighted, edge from 1 to n is the *only* shortest path.
   - We only need to make sure 1 and n get the same color.
   - Always possible, except if $n = 2$ and $k = 1$.

Statistics at 4-hour mark: 67 submissions, 15 accepted, first after 01:39