K: Kaktus Kourse

Problem author: Leon van der Waal



Problem: Find the length of the k-th longest simple path in a kaktus graph.

K: Kaktus Kourse

Problem author: Leon van der Waal



Problem: Find the length of the k-th longest simple path in a kaktus graph.

Naive solution: Loop over all starting points, and brute-force all possible paths using DFS.

K: Kaktus Kourse

Problem author: Leon van der Waal



Problem: Find the length of the k-th longest simple path in a kaktus graph.

Naive solution: Loop over all starting points, and brute-force all possible paths using DFS.

Verdict: **TLE**, but what is the complexity?

K: Kaktus Kourse

Problem author: Leon van der Waal

Problem: Find the length of the k-th longest simple path in a kaktus graph.

Naive solution: Loop over all starting points, and brute-force all possible paths using DFS.

Verdict: **TLE**, but what is the complexity?



Observation: Fix the first and last node of a simple path. There are 2^c possible simple paths between these nodes, where c is the number of cycles inbetween these nodes.

K: Kaktus Kourse

Problem author: Leon van der Waal



Observation: Fix the first and last node of a simple path. There are 2^c possible simple paths between these nodes, where c is the number of cycles inbetween these nodes.

Observation: c is at most $\left\lfloor \frac{m}{3} \right\rfloor$, since each cycle has at least 3 edges.

K: Kaktus Kourse

Problem author: Leon van der Waal



Observation: Fix the first and last node of a simple path. There are 2^c possible simple paths between these nodes, where c is the number of cycles inbetween these nodes.

Observation: c is at most $\left\lfloor \frac{m}{3} \right\rfloor$, since each cycle has at least 3 edges.

Observation: The worst case has $\left\lfloor \frac{m}{3} \right\rfloor$ cycles in a line.

K: Kaktus Kourse

Problem author: Leon van der Waal



Observation: Fix the first and last node of a simple path. There are 2^c possible simple paths between these nodes, where c is the number of cycles inbetween these nodes.

Observation: c is at most $\left\lfloor \frac{m}{3} \right\rfloor$, since each cycle has at least 3 edges.

Observation: The worst case has $\left\lfloor \frac{m}{3} \right\rfloor$ cycles in a line.

Observation: By geometric sums, there can only be $\mathcal{O}\left(2^{\frac{m}{3}}\right)$ simple paths.

K: Kaktus Kourse

Problem author: Leon van der Waal



Observation: Fix the first and last node of a simple path. There are 2^c possible simple paths between these nodes, where c is the number of cycles inbetween these nodes.

Observation: c is at most $\left\lfloor \frac{m}{3} \right\rfloor$, since each cycle has at least 3 edges.

Observation: The worst case has $\left\lfloor \frac{m}{3} \right\rfloor$ cycles in a line.

Observation: By geometric sums, there can only be $\mathcal{O}\left(2^{\frac{m}{3}}\right)$ simple paths.

Running time: Finding all possible simple paths is too slow, we need to do something more efficient.



Better solution: Do binary search on the answer. This way, we have to count the number of simple paths of length $\leq B$, when we binary search on B.

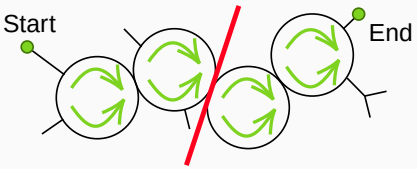


Better solution: Do binary search on the answer. This way, we have to count the number of simple paths of length $\leq B$, when we binary search on B.

For all start and end nodes, let there be k cycles inbetween the start and end node.

Better solution: Do binary search on the answer. This way, we have to count the number of simple paths of length $\leq B$, when we binary search on B.

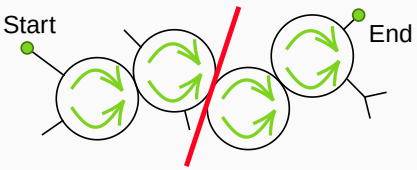
For all start and end nodes, let there be k cycles inbetween the start and end node.



Separate the first $\left\lceil \frac{k}{2} \right\rceil$ and last $\left\lfloor \frac{k}{2} \right\rfloor$ cycles.

Better solution: Do binary search on the answer. This way, we have to count the number of simple paths of length $\leq B$, when we binary search on B.

For all start and end nodes, let there be k cycles inbetween the start and end node.

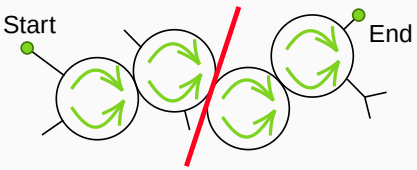


Separate the first $\left\lceil \frac{k}{2} \right\rceil$ and last $\left\lfloor \frac{k}{2} \right\rfloor$ cycles.

For both halves, precompute and sort all possible path lengths in $\mathcal{O}\left(2^{\frac{k}{2}}\right)$ time.

Better solution: Do binary search on the answer. This way, we have to count the number of simple paths of length $\leq B$, when we binary search on B.

For all start and end nodes, let there be k cycles inbetween the start and end node.



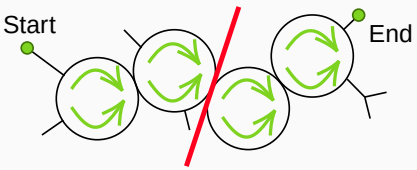
Separate the first $\left\lceil \frac{k}{2} \right\rceil$ and last $\left\lfloor \frac{k}{2} \right\rfloor$ cycles.

For both halves, precompute and sort all possible path lengths in $\mathcal{O}\left(2^{\frac{k}{2}}\right)$ time.

For a binary search bound B, we can count the number of simple paths in $\mathcal{O}\left(2^{\frac{k}{2}}\right)$, using two pointers.

Better solution: Do binary search on the answer. This way, we have to count the number of simple paths of length $\leq B$, when we binary search on B.

For all start and end nodes, let there be k cycles inbetween the start and end node.



Separate the first $\left\lceil \frac{k}{2} \right\rceil$ and last $\left\lfloor \frac{k}{2} \right\rfloor$ cycles.

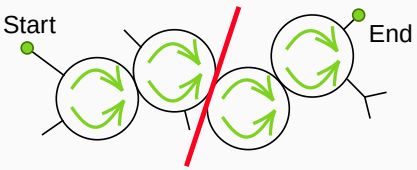
For both halves, precompute and sort all possible path lengths in $\mathcal{O}\left(2^{\frac{k}{2}}\right)$ time.

For a binary search bound B, we can count the number of simple paths in $\mathcal{O}\left(2^{\frac{k}{2}}\right)$, using two pointers.

Running time: By geometric sums, the overall complexity is hence $\mathcal{O}\left(2^{\frac{m}{6}} \log(nA)\right)$, with $A = 10^9$.

Better solution: Do binary search on the answer. This way, we have to count the number of simple paths of length $\leq B$, when we binary search on B.

For all start and end nodes, let there be k cycles inbetween the start and end node.



Separate the first $\left\lceil \frac{k}{2} \right\rceil$ and last $\left\lfloor \frac{k}{2} \right\rfloor$ cycles.

For both halves, precompute and sort all possible path lengths in $\mathcal{O}\left(2^{\frac{k}{2}}\right)$ time.

For a binary search bound B, we can count the number of simple paths in $\mathcal{O}\left(2^{\frac{k}{2}}\right)$, using two pointers.

Running time: By geometric sums, the overall complexity is hence $\mathcal{O}\left(2^{\frac{m}{6}} \log(nA)\right)$, with $A = 10^9$.

Statistics: 4 submissions, 0 accepted