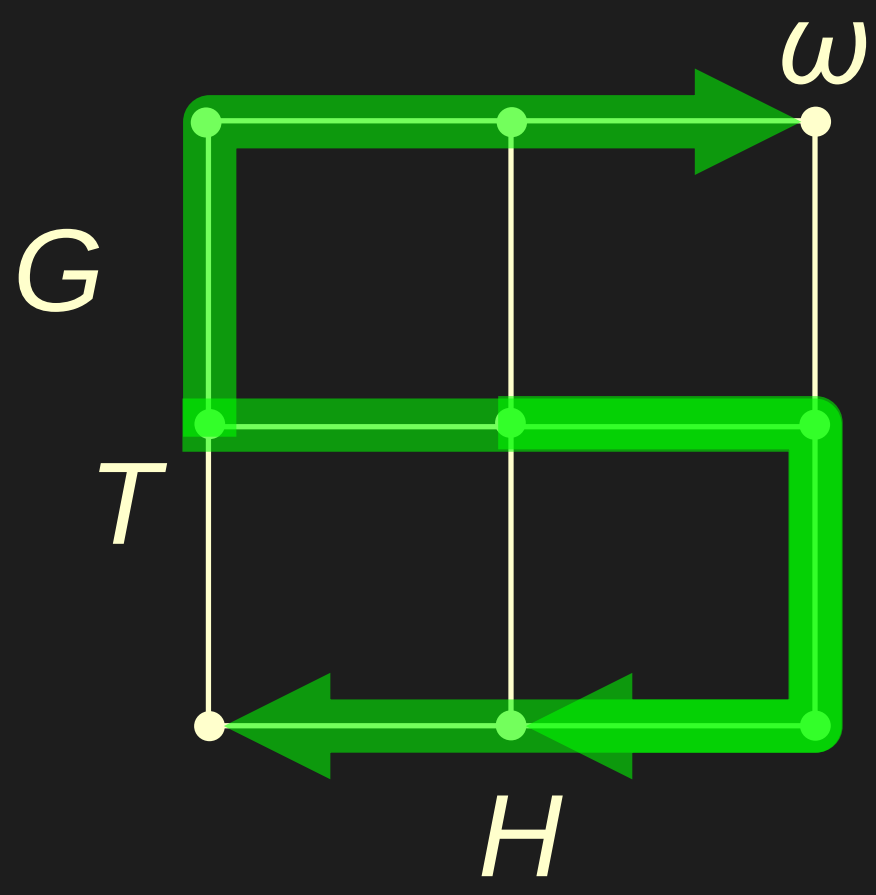
Minko Markov, Sofia University, *minkom@fmi.uni-sofia.bg*
Krassimir Manev, New Bulgarian University, *kmanev@nbu.bg*

The TRAIN IN A GRAPH *task*: Intractability in General case and an $O(n)$ Algorithm on Cacti

Preliminaries



Graph *G*, a *train* with *head* in *H* and *tail* in *T* and a *target vertex ω*.

Step – moving the head in free vertex

Transformation of train to *ω*

Preliminaries

- Pankov, P. (2008). Naturalness in tasks for olympiads in informatics. *Olympiads in Informatics*, v. 2,115–121 – transform train to T but tail to be in H, called *reversing*
- Torii, R. (2008). Path transferability of graphs. *Discrete Mathematics*, 308(17), 3782–3804 – necessary and sufficient condition for *G* in order that each given train to be transformable (or reversible)

Definition of the problem

- *A train* in an undirected graph $G = (V, E)$ is a simple path $p = x_0, x_1, \ldots, x_q$ of length q, x_0 is *the head*. *The free vertices* with respect to p are $V \setminus \{ x_0, x_1, \ldots, x_q \}$.
- *A step* of the train is a transformation $p \rightarrow p'$ where p' is a simple path $p' = x', x_0, x_1, \ldots, x_{q-1}$, where x' is a free vertex with respect to p, $(x', x_0) \in E$.

Definition of the problem (2)

- After the move, the train is p' and the head is in x'. The free vertices with respect to p' are $V \setminus \{ x', x_0, x_1, \ldots, x_{q-1} \}$.
- The length of the train is constant.
- *The target* is some vertex $\omega \in V$. The target never changes during the transformation.
- Without loss of generality, assume G is connected.

TIAG

We call this task TRAIN IN A GRAPH, shortly TIAG.

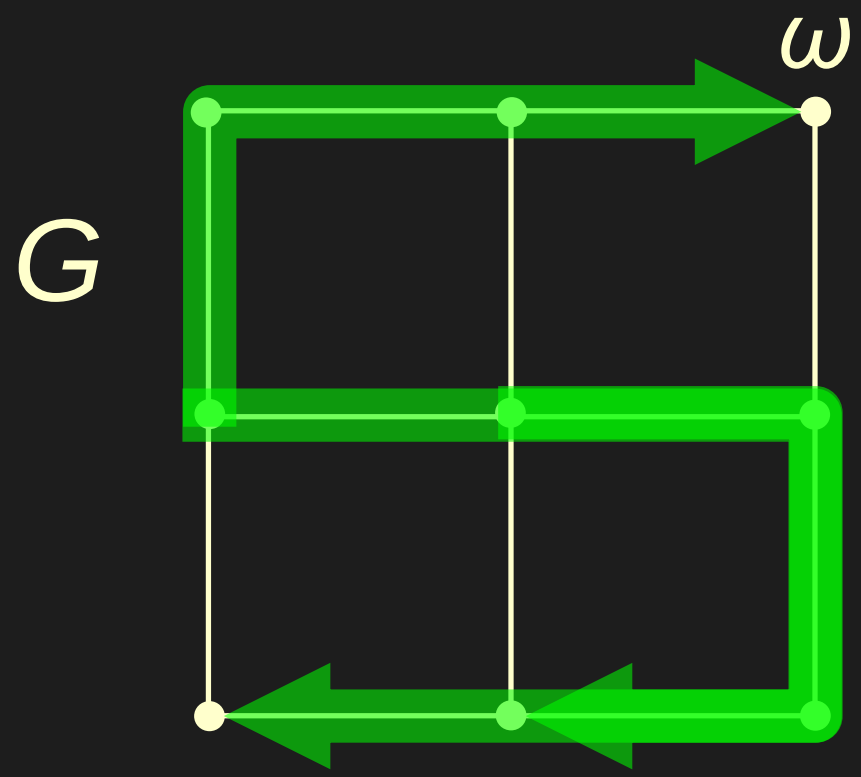
Decision version

- generic instance: undirected connected graph G, a train p in it, and a target, ω
- question: can the locomotive reach the target.

TIAG, optimization version

- generic instance: undirected connected graph G, a train p in it, and a target ω
- output: the minimum number of moves after which locomotive can reach the target, in case that is possible, otherwise some indication the target is not reachable.

A YES-instance of TIAG



General considerations

- The length of the train can be zero but in that case the answer to the decision version is trivially YES and the optimization version is solved by BFS.
- The only reason the train cannot commence immediately a travel along a shortest path towards the target is that its “body” blocks all such paths; recall that the train cannot intersect itself.

General considerations (2)

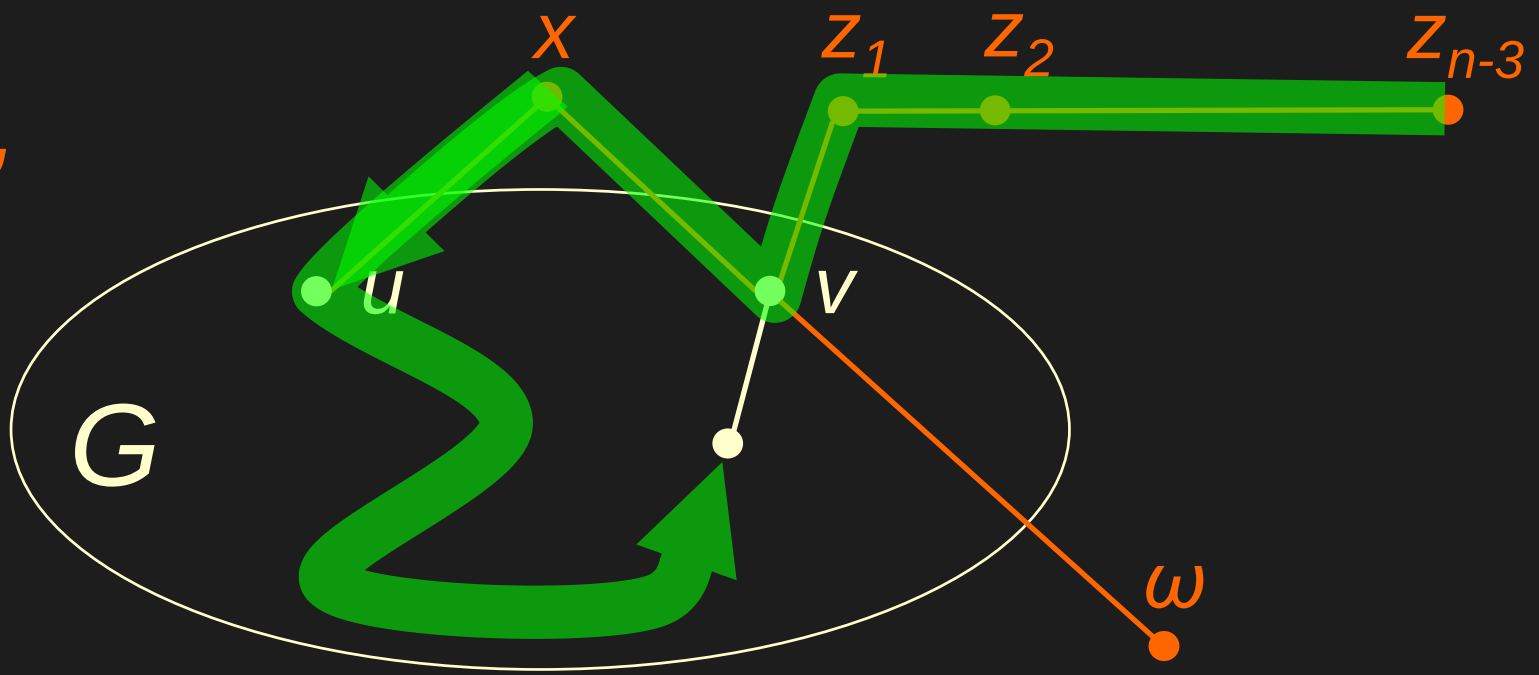
- Ostensibly, that is a minimization problem.
- In fact, the problem is a maximization one: in the worst case, in order to get the head to the target, we have to find a long enough path to “store” the train’s body and thus clear a path to the target.
- Computing long paths is hard in general.

Intractability results

- Hamiltonian Path between Two Points (HPBTP):
  - generic instance: Undirected graph *G*, vertices *u* and *v* in *G.*
  - question: Does there exist a Hamiltonian path with endpoints *u* and *v* in *G*?
- HPBTP is *NP*-complete (Garey and Johnson).
- HPBTP reduces to TIAG.

HPBTP $\leq_p$ TIAG

No path in G can be longer than $n-1$. If v is to be free at that moment, the part of the train inside G has to have length $n-2$. That implies H-path between u and v.

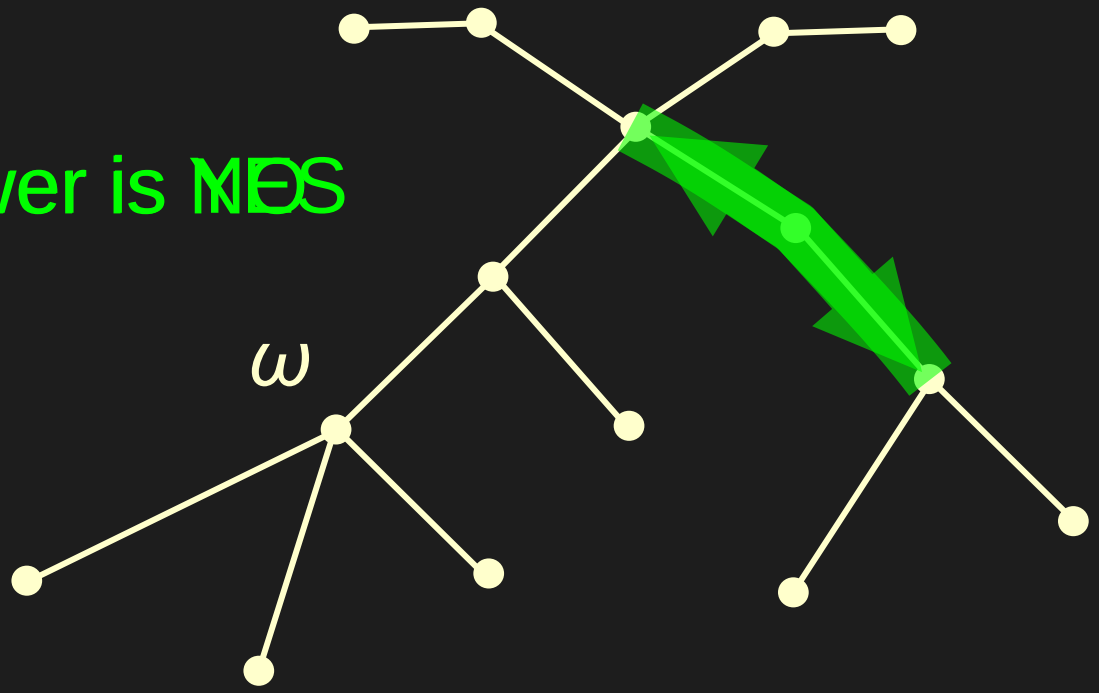


NP-completeness of TIAG

- So far we have proved *NP*-hardness.
- The fact that TIAG $\in$ *NP* is not immediately obvious.
- We prove that for every YES-instance the locomotive moves along a path that does not contain any vertex more than twice.
- The fact that a succinct certificate exists follows immediately.

TIAG on trees is trivial

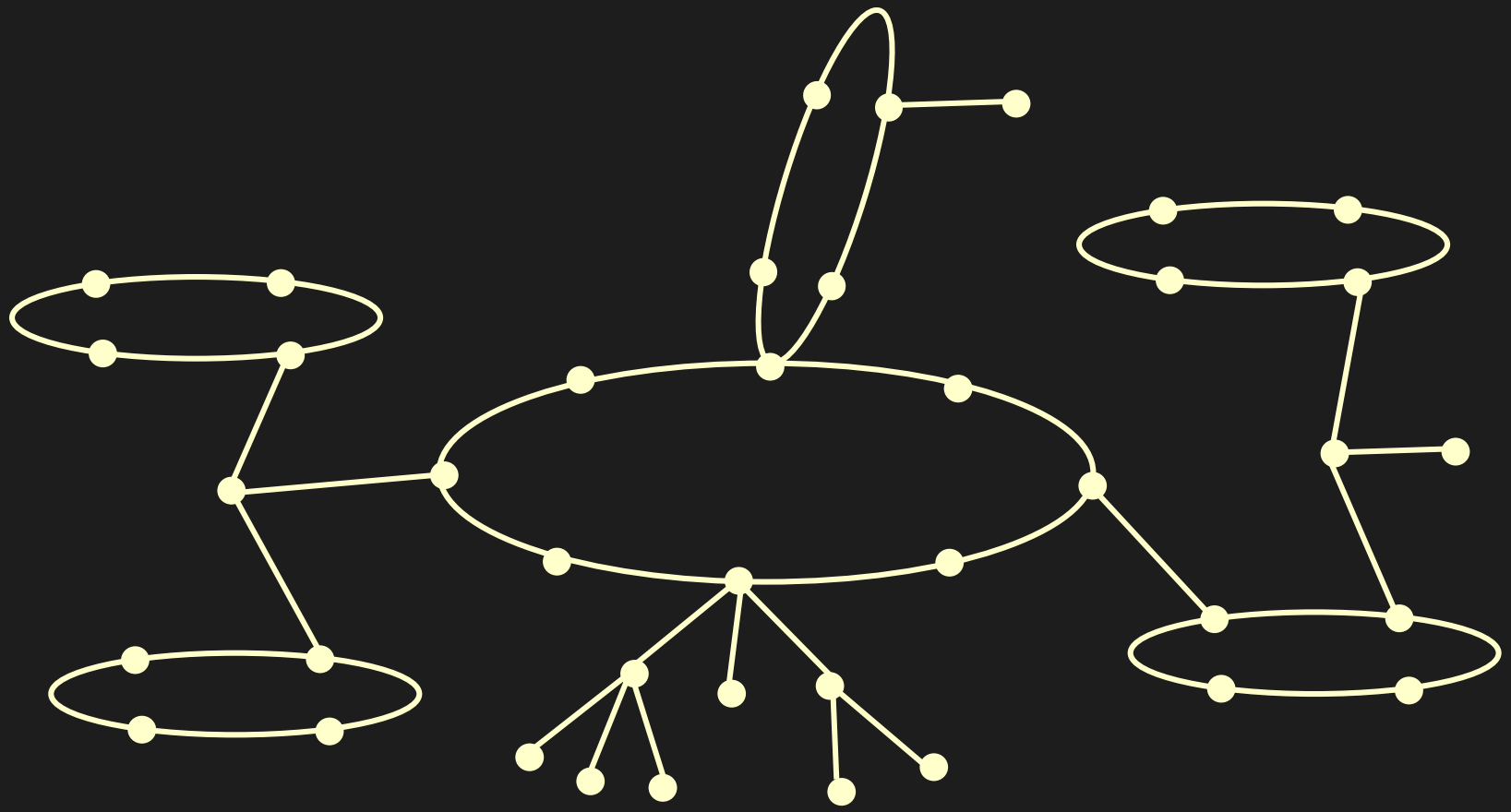
Answer is YES



TIAG on other graphs

- Reversing the train in a complete graph K_n - XV Akademickie Mistrzostwa Polski w Programowaniu Zespołowym, 2010
- Transforming the train to arbitrary vertex in a "*vertex-cacti*" – as well as we know that the task was included in one of the Russian Open Cup programming contest ???
- We consider the same task in a "*edge-cacti*" called simply *cacti.*

Cactus Graphs (cacti)

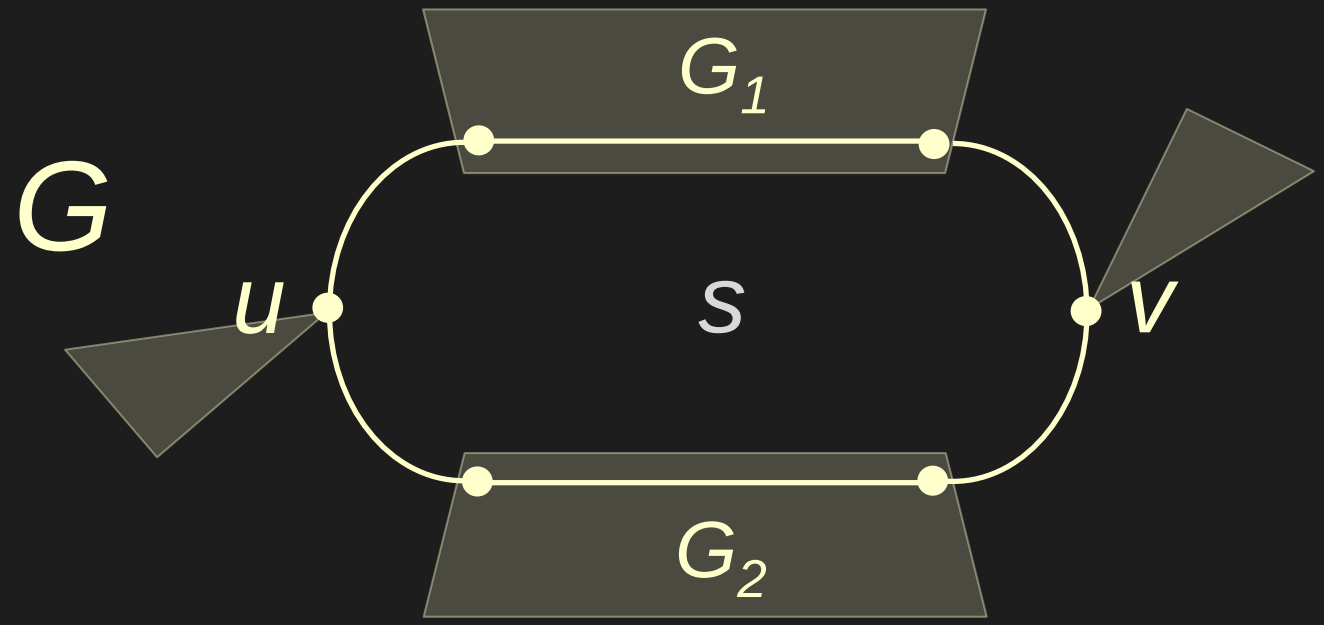


Krassimir Manev, New Bulgarian University

Definitions on Cacti (1)

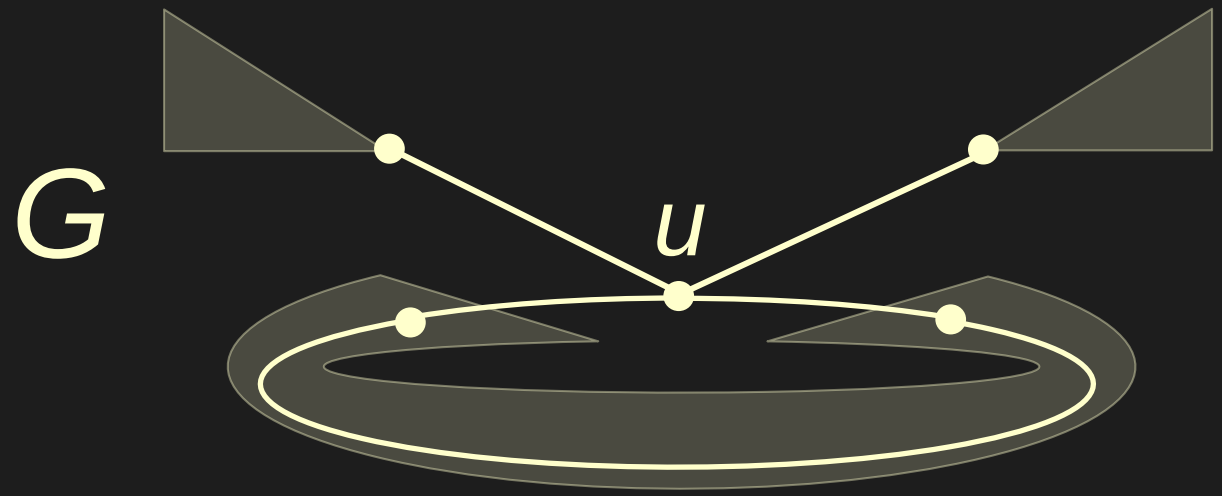
the cacti G_1 and G_2 are *the constituents* of s with respect to u and v

s is a cycle, u and v are vertices in it.



Definitions on Cacti (2)

the fragments of G with respect to vertex *u* are the connected components of *G*−*u*



Definitions on Cacti (3)

- With respect to the train's length *q*, any cycle of length ≥ *q+1* is *a long cycle*.
- If there are no long cycles, the problem is as trivial as it is on trees: the answer is YES iff there is a free path between the head and *ω* initially.
- If there are long cycles, they are called *U-turns* of the train, giving a lot more opportunities.

Definitions on Cacti (4)

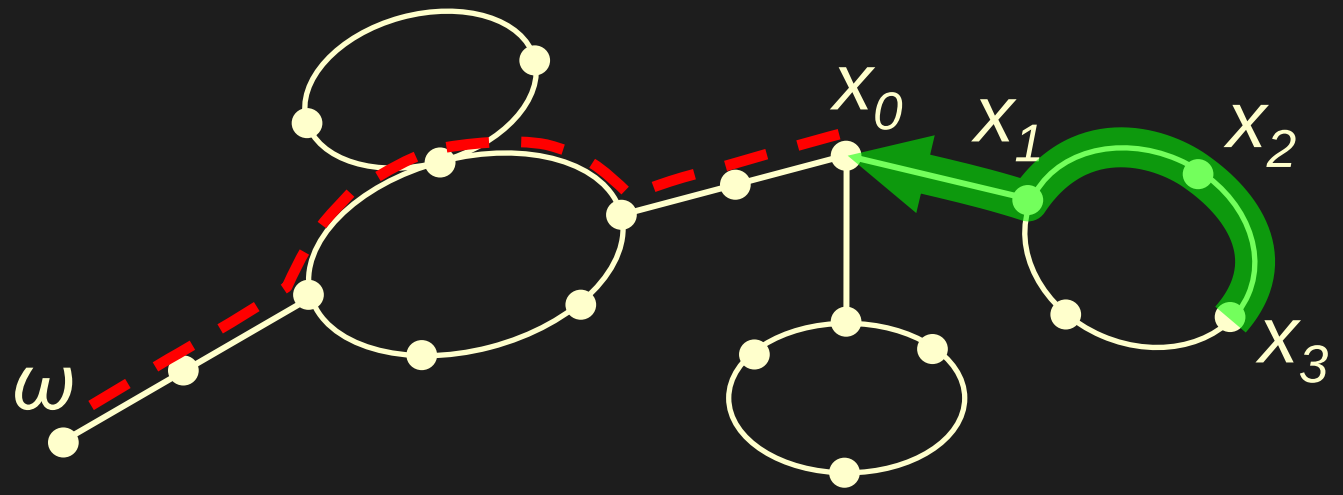
- The edges of a cactus are partitioned into *tree edges* and *cycle edges* in the obvious way.

TIAG on Cacti (1)

- We solve the optimization version of TIAG in *O(n)* on cacti.
- Let the cactus be *G* and the train be $p = x_0, x_1, \ldots, x_q$.

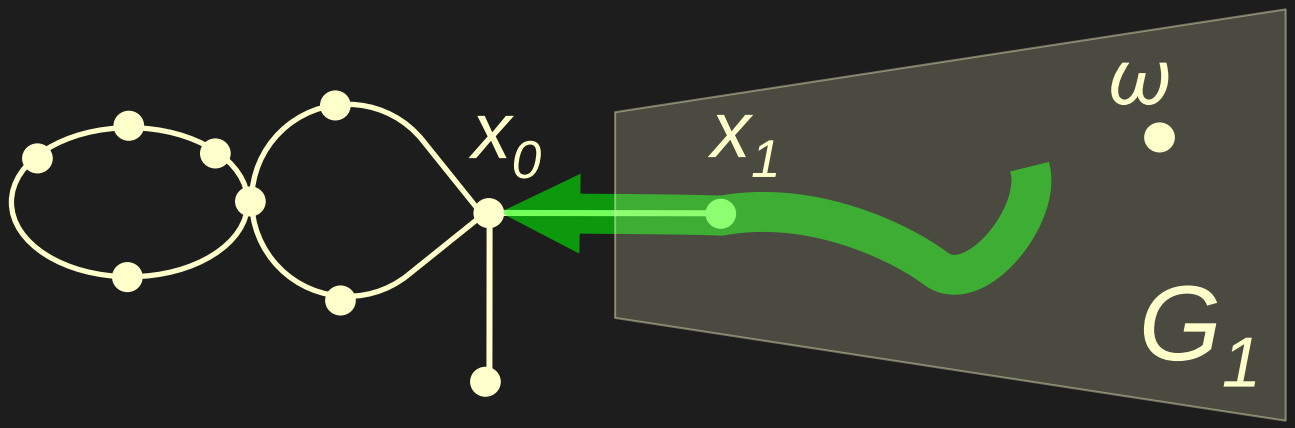
TIAG on Cacti (2)

- If x_1 and ω are not in the same fragment with respect to x_0 the answer is, the length of a shortest path between x_0 and ω.



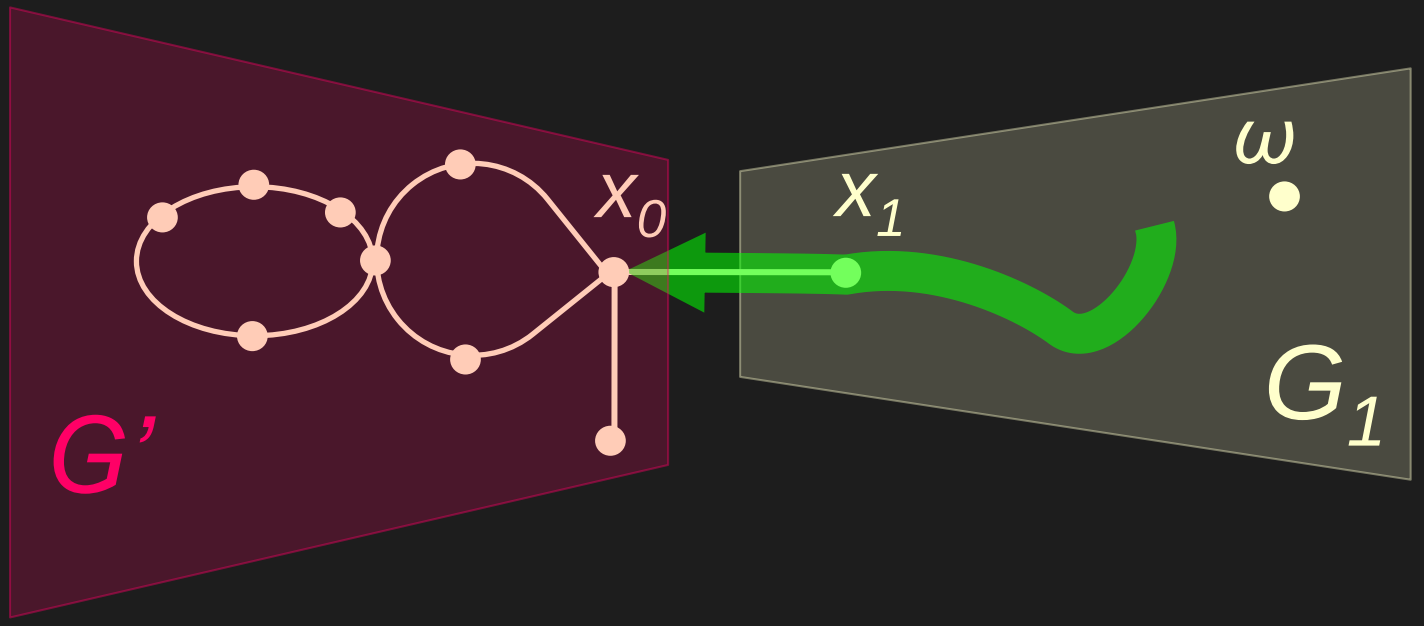
TIAG on Cacti (3)

- The interesting case is if x_1 and ω being in the same fragment G_1 with respect to x_0.



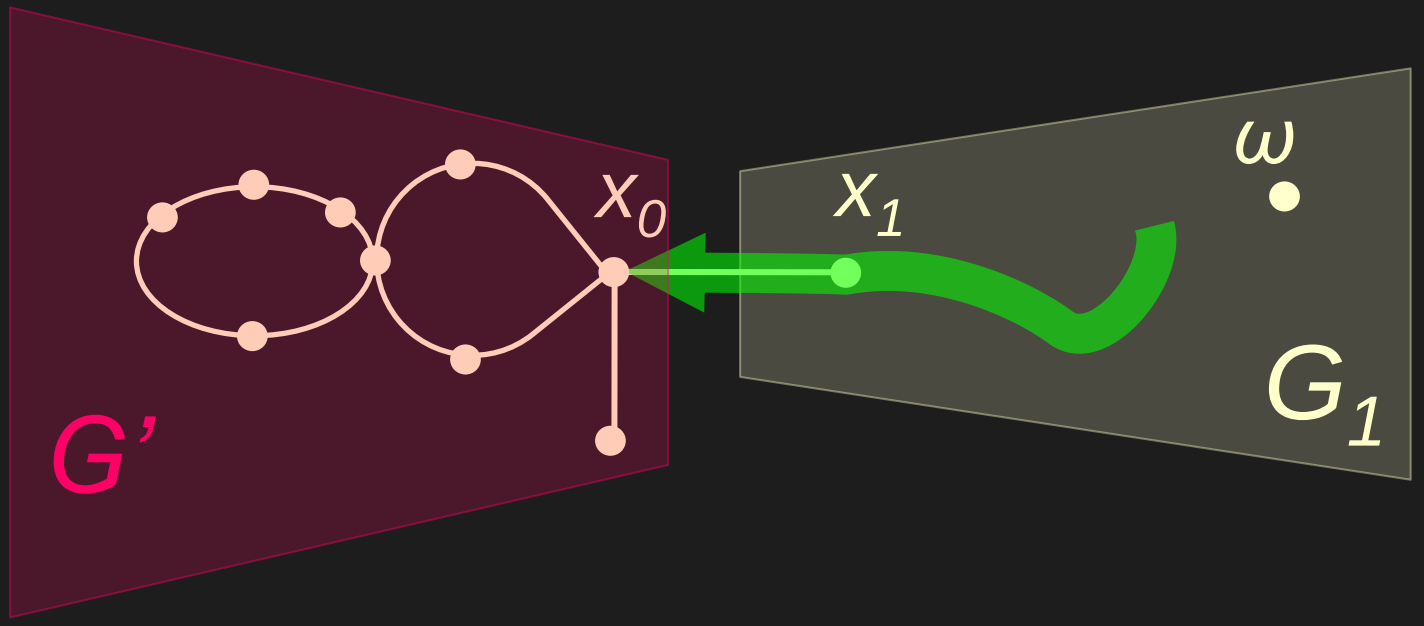
Subcase 1: (x_0, x_1) is a tree edge

- Let G' be the connected component of $G - (x_0, x_1)$ that does not contain x_1.



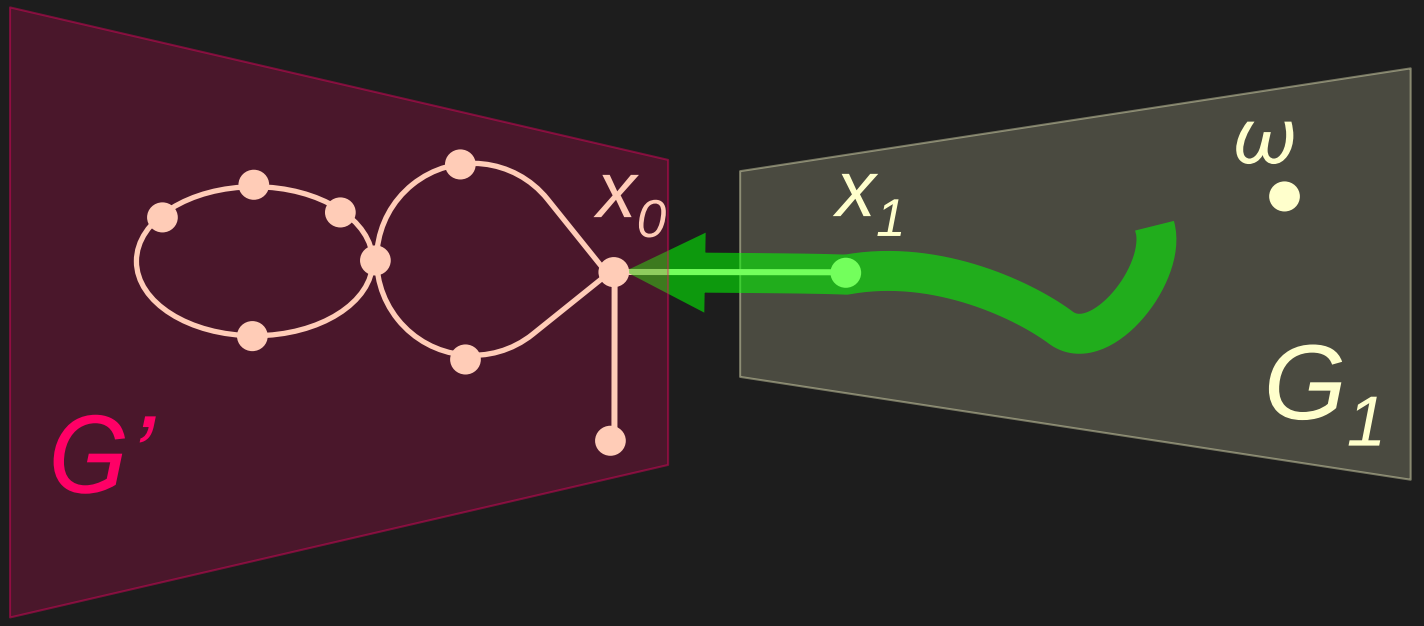
Subcase 1: (x_0, x_1) is a tree edge

- If there is no U-turn for the train inside G', the answer is ∞.



Subcase 1: (x_0, x_1) is a tree edge

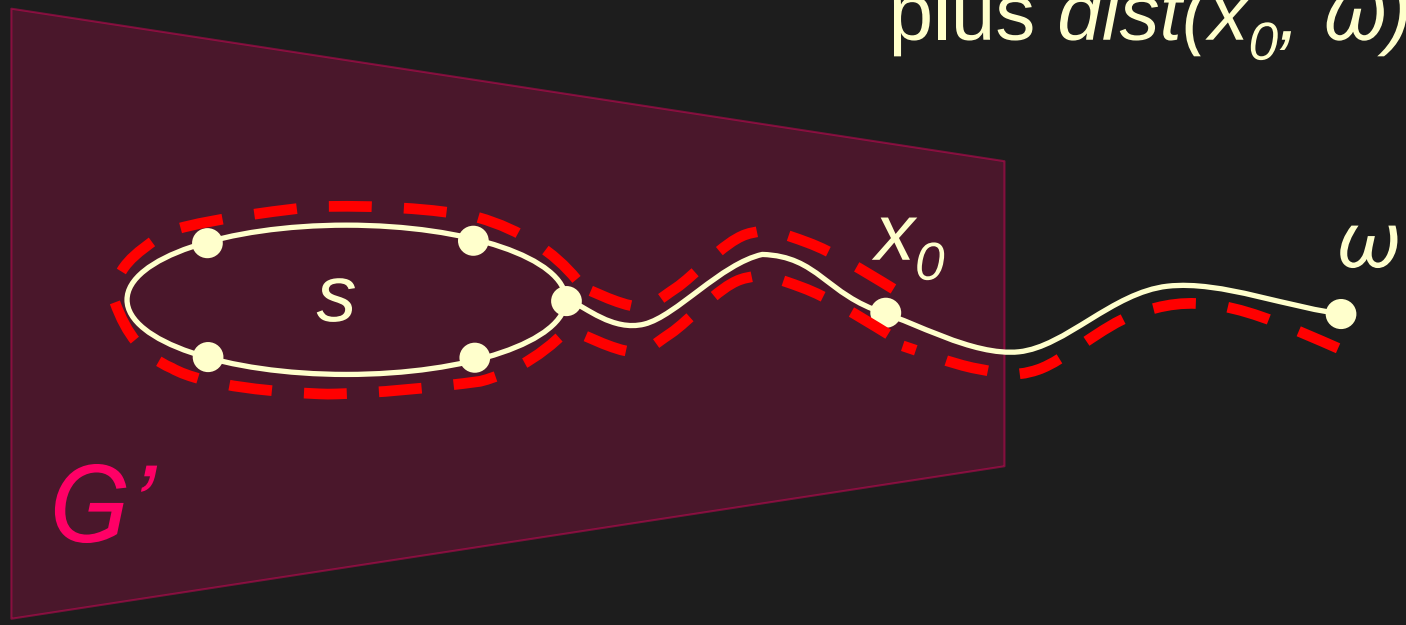
- Else, the answer is the shortest U-turn inside G' plus the distance between x_0 and ω.



Subcase 1: (x_0, x_1) is a tree edge

the length of the U-turn

plus $dist(x_0, \omega)$



Subcase 1: (x_0, x_1) is a tree edge

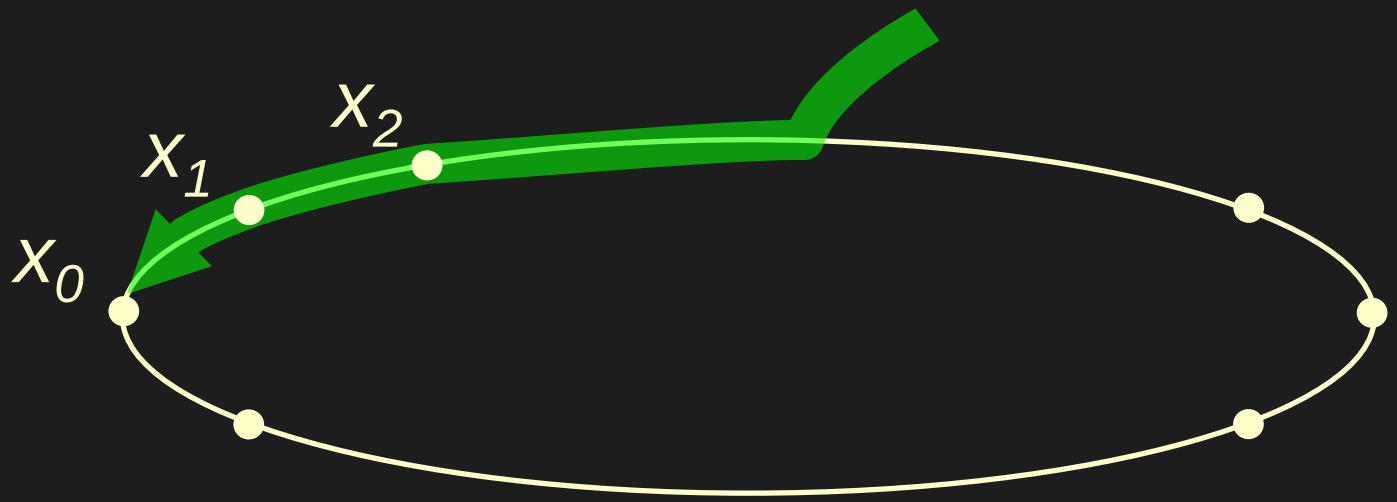
- There can be multiple opportunities to take a similar U-turn inside *G'*
- So we take the minimum over all long cycles in *G'* of:
  - the cycle's length plus
  - twice the distance between the cycle and vertex x_0.

Subcase 1: (x_0, x_1) is a tree edge

- That information can be precomputed: a modified DFS can discover in a single run, starting at x_0, all long cycles and their distances to x_0.
- Combination of one DFS and one BFS could make the implementation more easy

Subcase 2: (x_0, x_1) is a cycle edge

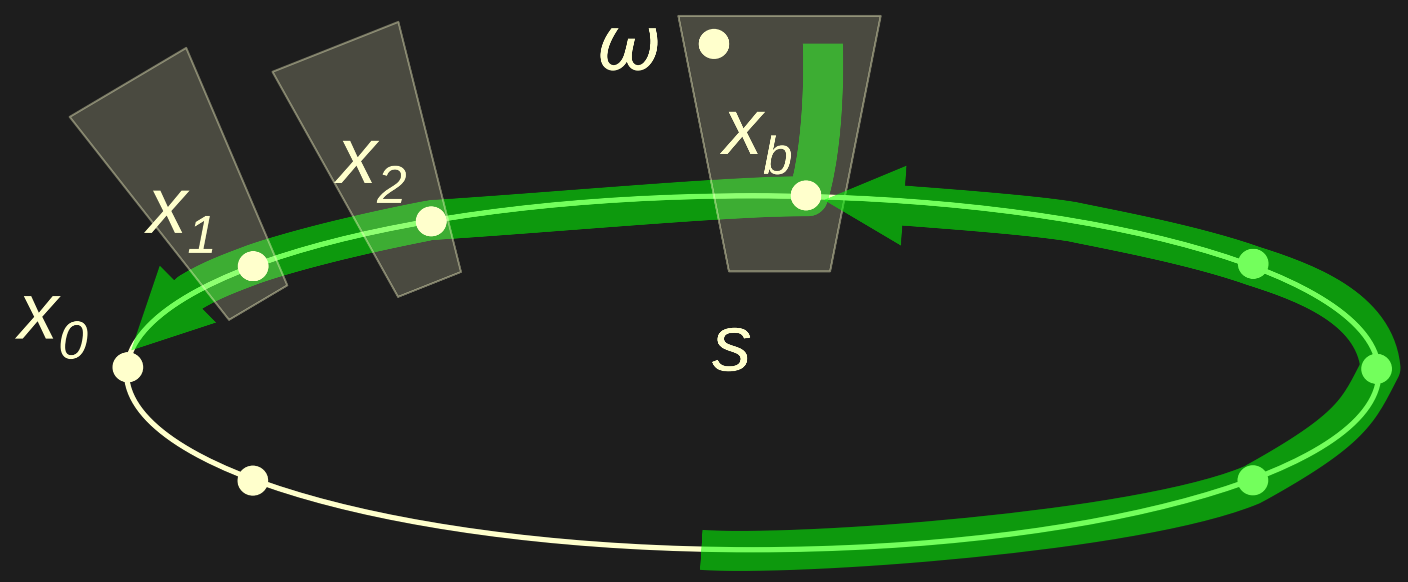
- Further subdivision into subcases is according to the relative placement of the train and ω:



Subcase 2.1: ω is “blocked"

it either can go directly to ω along s, provided s is a long cycle (so it can “accommodate” the train)

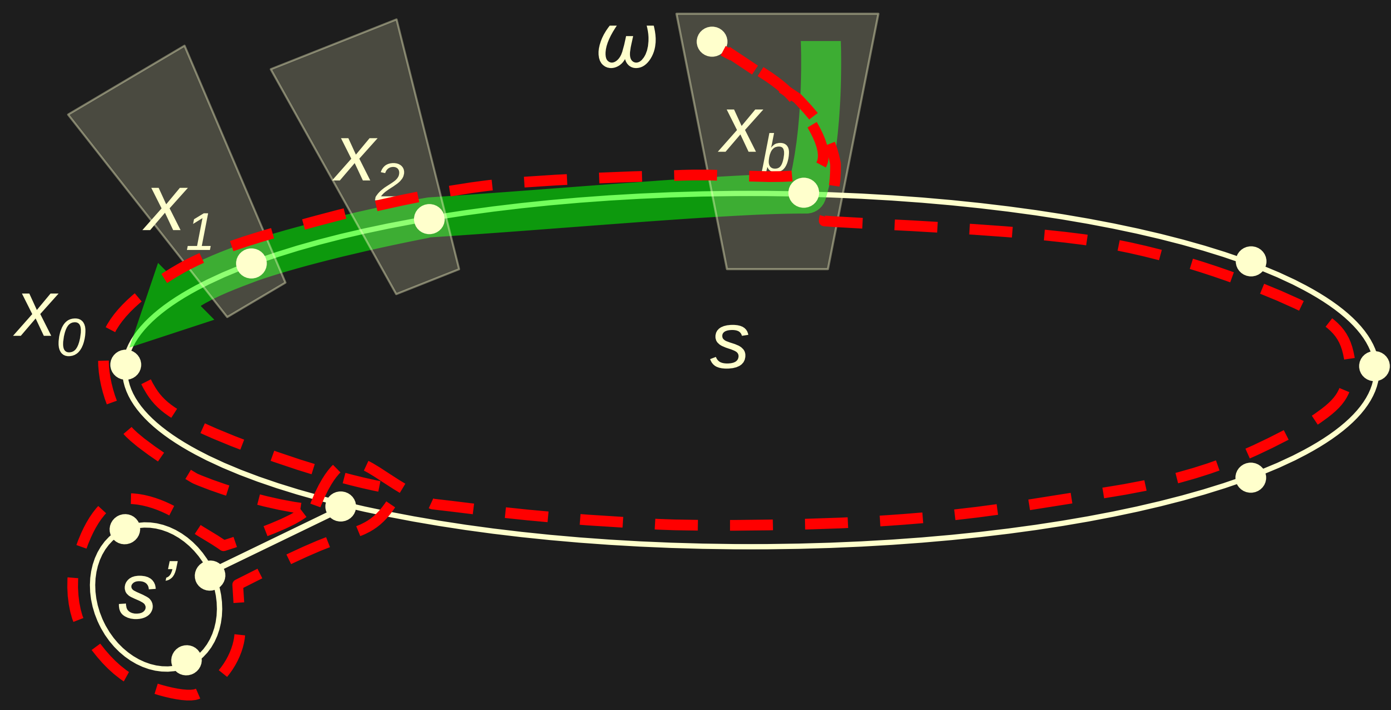
if it is in a conduit of s below x_1 or x_b the train has to move forward in order to free a path to ω



Subcase 2.1: ω is “blocked"

after the train can go “backwards” only if s is not below $x_1 \ldots x_b$

or go “forward” along the long cycle

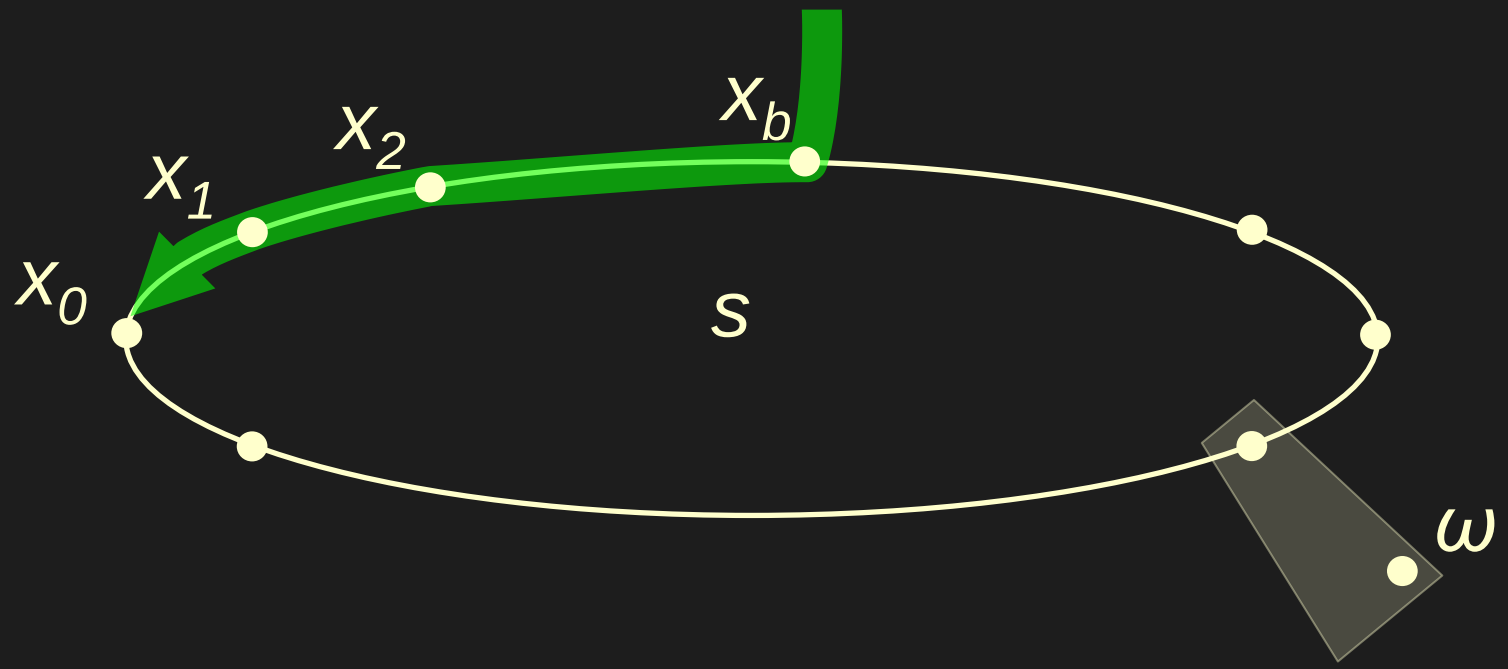


Subcase 2.1: ω is “blocked"

- So, there are at most 3 possibilities:
  - do not make a U-turn,
  - make a U-turn and go backwards.
  - make a U-turn and go forward.
- If a suitable precomputing is done, the minimum of all of them can be computed in a linear time
- The said modified DFS will do as a precomputing

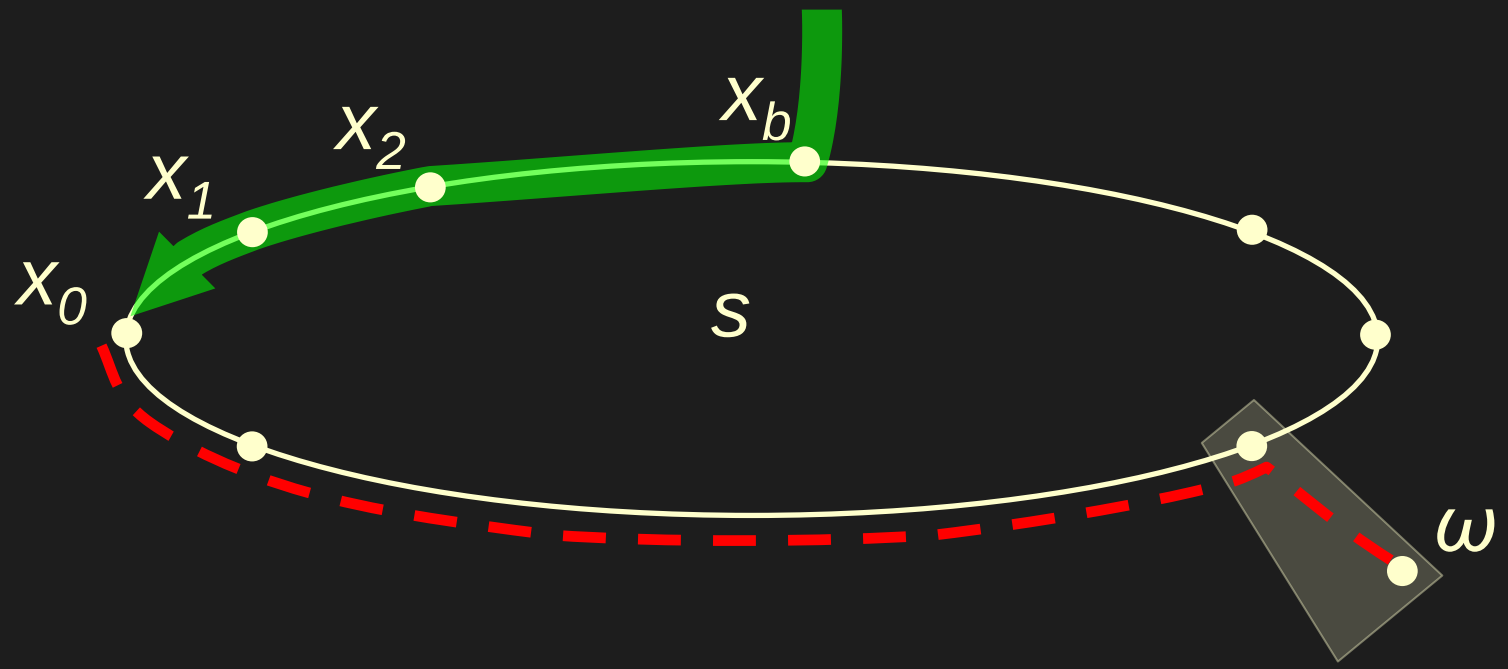
Subcase 2.2: ω is not “blocked"

- Now there is necessarily a solution to the decision problem



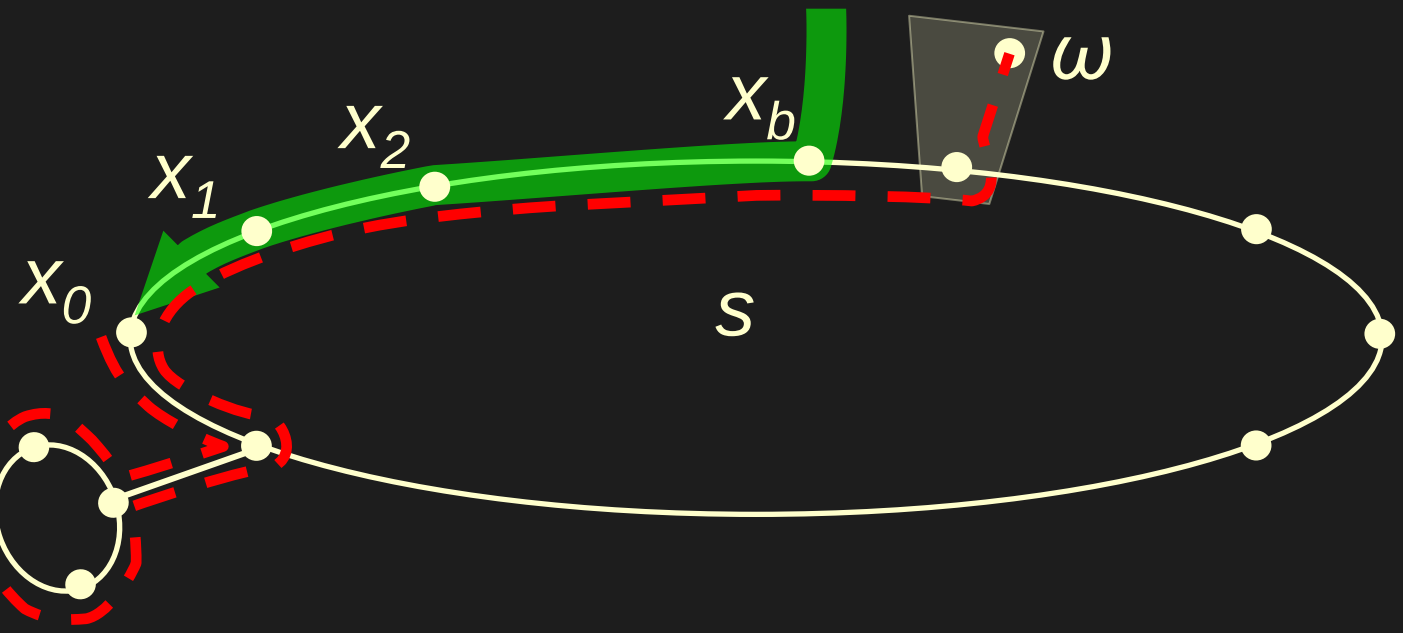
Subcase 2.2: ω is not “blocked"

- However, the solution to the optimization problem is not necessarily the obvious one



Subcase 2.2: ω is not “blocked"

- It can be more beneficial to perform a U-turn and then go backwards, if s is long enough and the train is short enough and there is a suitable U-turn



Subcase 2.2: ω is not “blocked"

- The minimum of those two possibilities can also be computed in linear time using the results of the mentioned pre-computing
- That exhausts all possibilities!

Conclusions

- We have proved the NP-completeness of an interesting computational problem that has only been mentioned so far
- We have constructed a linear-time algorithm for it on cacti. The implementation details are not as straightforward as they sound: the modified DFS (or DFS+BFS combination) is tricky.

Conclusions (2)

- There are plenty of opportunities for further research:
  - There are numerous other graph classes to consider, for instance, planar graphs. Many NP-complete problems remain so on planar graph. What about TIAG?
  - The train movement can be made more complicated

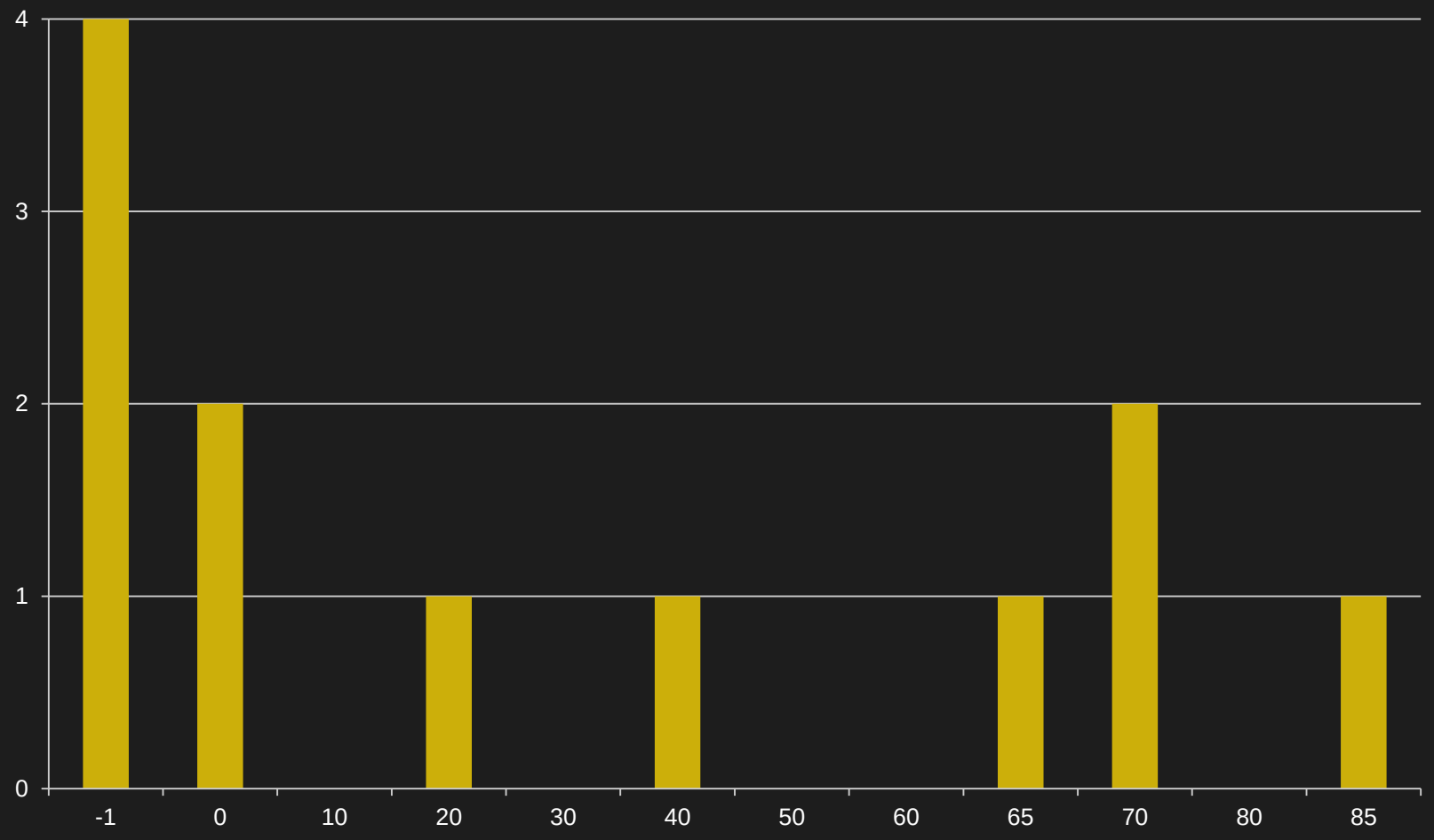
Conclusions (3)

- Task was used in a contest for choosing the National team of Bulgaria in May 2013
- 12 contestants took part
- Data was constrained as follow:

  $2 \le n \le 100000, 1 \le q < N$

- The results are shown on the diagram of the next slide and demonstrate that the task is difficult even on cacti

Conclusions (4)



Thank you for your attention!

Any questions?