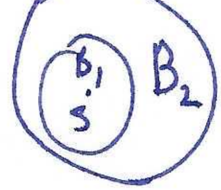
B_i = distance-i ball from s

$= \{v \mid d(s,v) \leq i\}$

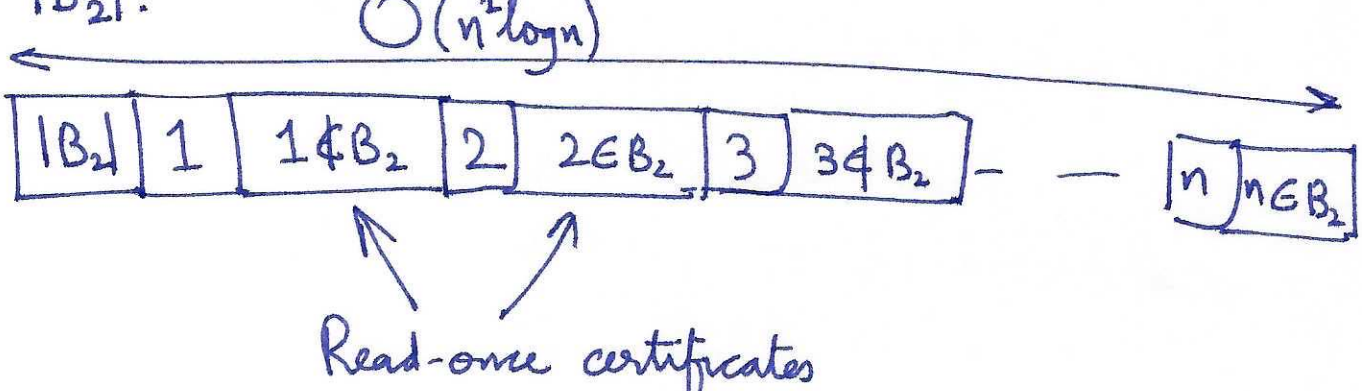
$B_0 = \{s\}$ $\quad$ $B_1 = \Gamma^+(s)$



* There is a read-once certificate of length $O(n\log n)$ for v s.t. $d(s,v) > 2$ ($v \notin B_2$)

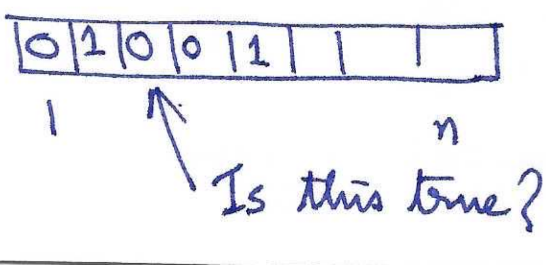
* There is a read-once cert. of length $O(\cancel{n}\log n)$ for v s.t. $d(s,v) \leq 2$. (Just the path of length ≤ 2)
($v \in B_2$)

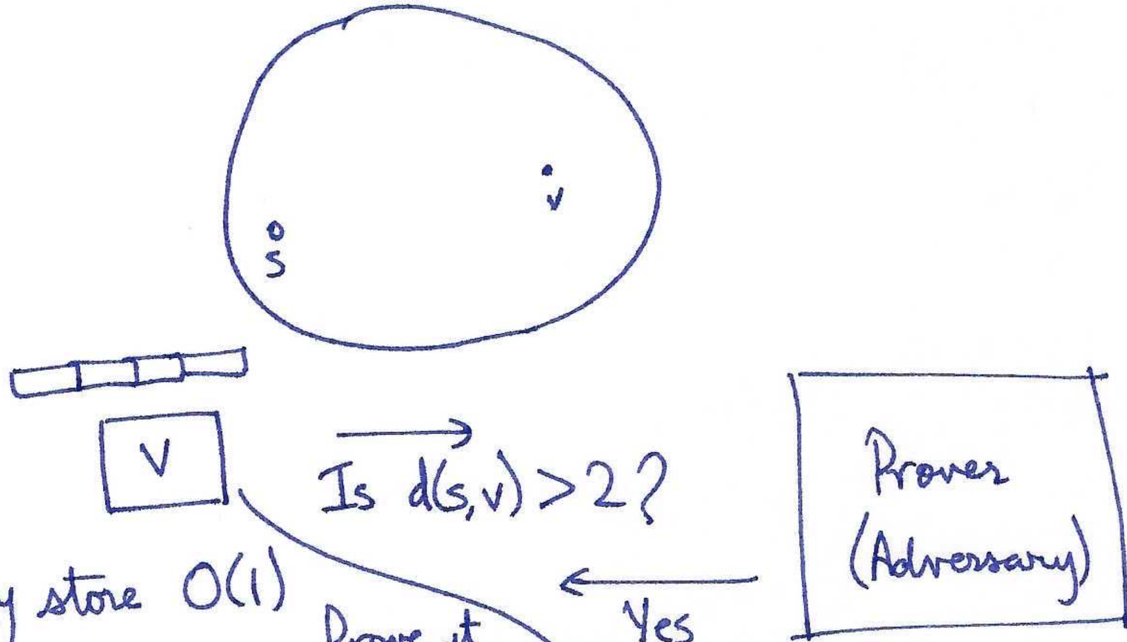
We can construct a read-once certificate for the size $|B_2|$.



Verifier stores the certificate's "claimed" $|B_2|$.

For each vertex v in order, verifier get a certificate for $v \in B_2$ or $v \notin B_2$. Verifier keeps track of the number of vertices in B_2, and checks this is the same as $|B_2|$.





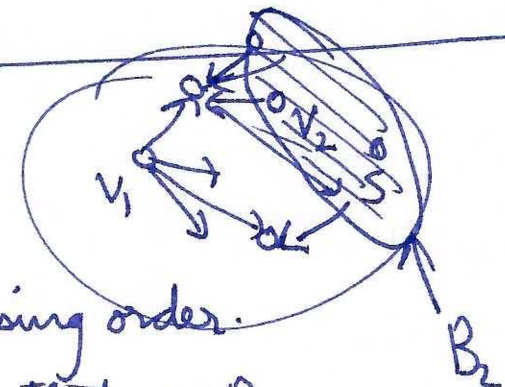
V can only store $O(1)$ vertices / numbers

$\longleftrightarrow$ $O(\log n)$

Prove it $\longrightarrow$ using a read-once certificate

$\longleftrightarrow$ poly(n)

V can look up the graph.

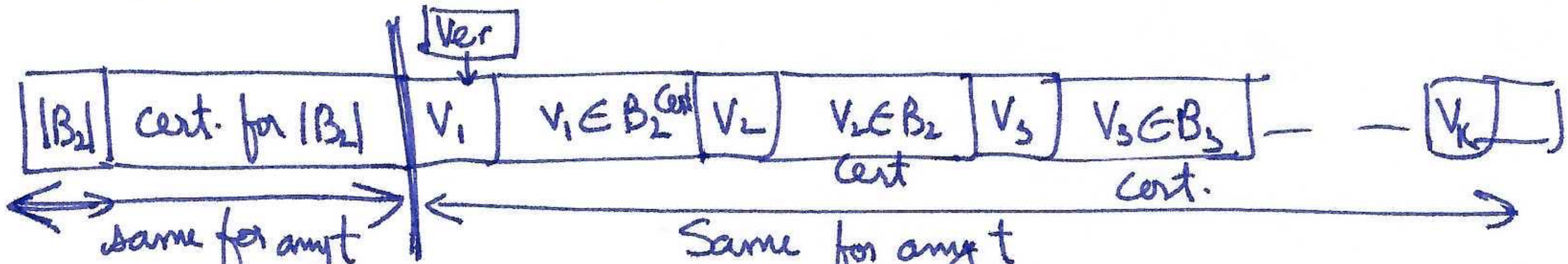


$t \notin B_3$ certificate

$|B_2|$

All vertices in B_2, in increasing order.

For each vertex v in B_2, cert. that $v \in B_2$.



Verifier (1) Verifies that $|B_2|$ is correct

(2) For each v_i in $v_1, \ldots v_k$

(a) Check that $v_i \in B_2$ is correct cert.

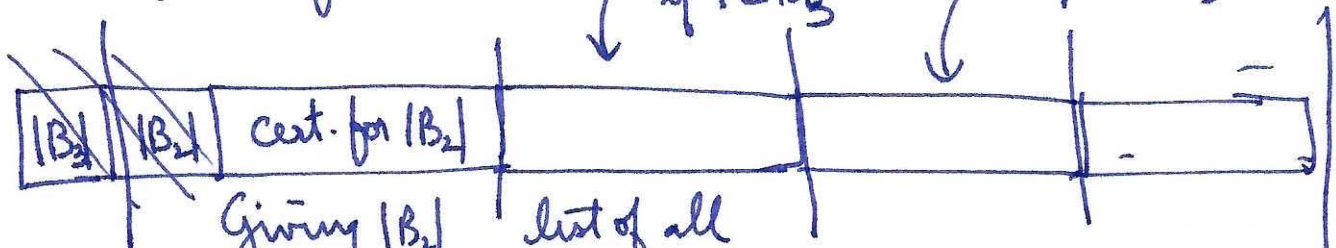
(b) Check that ~~$(v_i$~~ $(v_i, t) \notin E$

(3) Check that $k = |B_2|$

Certificate for $|B_3|$

Used to check if $1 \in B_3$

Used to check if $2 \in B_3$



Giving $|B_2|$

list of all vertices in $|B_2|$ with ~~size~~ certificate for each path

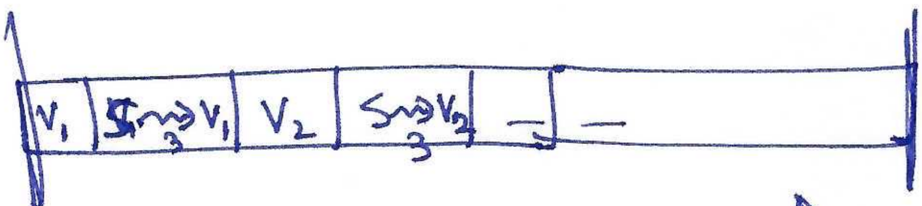
n times

$\rightarrow$ n paths of length n $O(n^2 \log n)$

$O(n^3 \log n)$

Verifier ~~is convinced~~ convinced of $|B_3|$

Cert size $= O(n^4 \log n)$



Each v_i in B_3, with a cert. that $v_i \in B_3$

$\underbrace{s \underset{3}{\rightsquigarrow} v_i}$

Path of length 3 from s to v_i

Cert. is enough to determine if $t \notin B_4$

Length $= O(n^2 \log n)$

If logspace machine can be ~~conv~~ convinced of $|B_i|$,

then given all vertices v of B_i in order, with a certificate of $v \in B_i$, machine can be convinced that $t \notin B_i$ (or not).

To ensure it has seen ALL vertices in B_i, it needs the size of $|B_i|$. By going over all vertices (and seeing all of B_i n times), it can count $|B_{i+1}|$.

Eventually, it can count $|B_{n-1}|$ = # reachable vertices.
This gives certificate that $t \notin B_{n-1}$, meaning $\langle G, s, t \rangle \notin$ PATH. ∎

Verifier knows $|B_i|$

Cert. Block: has each vertex v in B_i, followed by path from s to v of length $\leq i$

Size $= O(n^2 \log n)$ $\boxed{\text{Cert Block}}$

Given a vertex t and ONE read-once cert. block, verifier can determine if $t \notin B_{i+1}$ or $t \in B_{i+1}$.

With n read-once cert. blocks, verifier can count # vertices in B_{i+1}.

Total cert. Size $= O(n^3 \log n)$ to go from $|B_i|$ to $|B_{i+1}|$.

Overall, $n \times O(n^3 \log n) = O(n^4 \log n)$ to go from $|B_0| = 1$ to $|B_{n-1}|$