Streaming Algorithms

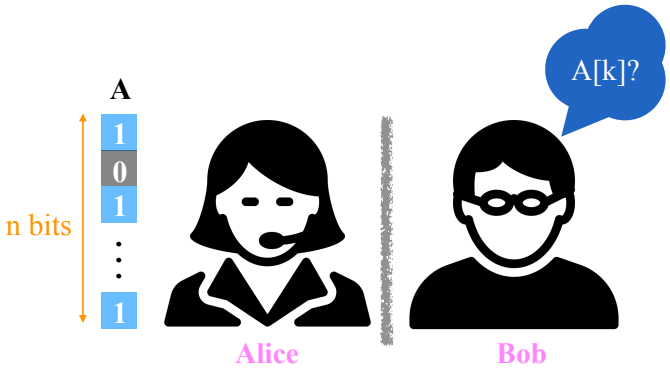
Meng-Tsung Tsai

05/15/2018

References

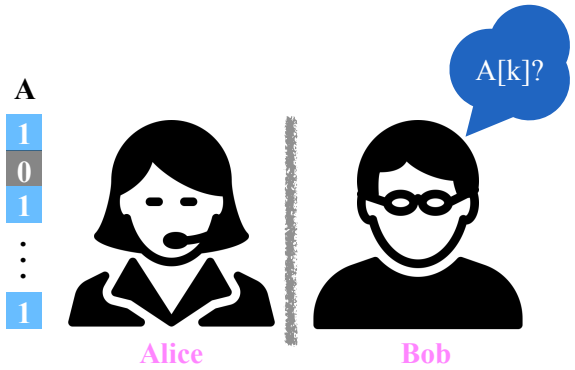
- "Communication Complexity," Kushilevitz
- "On graph problems in a semi-streaming model," Feigenbaum et al. (2005)
- "Vertex and Hyperedge Connectivity in Dynamic Graph Streams," Guha et al. (2015)

The indexing problem



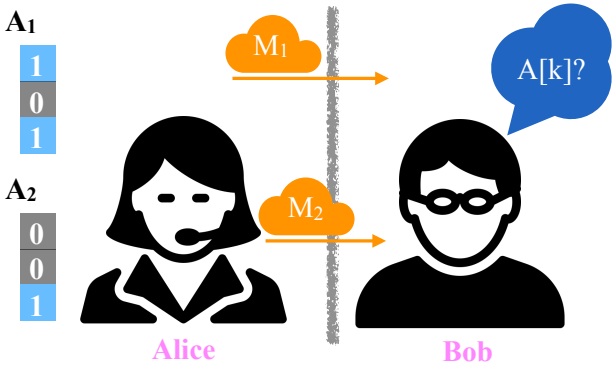
How many bits are needed to be sent from Alice to Bob so that Bob can figure out whether A[k] = 1 or not?

The indexing problem



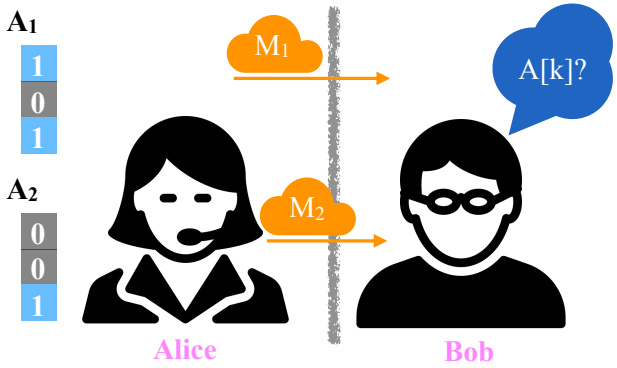
It needs Ω(n) bits for any protocol. (Why?)

The indexing problem



The messages M_1 and M_2 cannot be equal. Otherwise, let k = 1, then Bob has no way to distinguish the two cases.

The indexing problem



There are 2^n different n-bit array, and thus there are at least 2^n different messages, which requires n bits to represent.

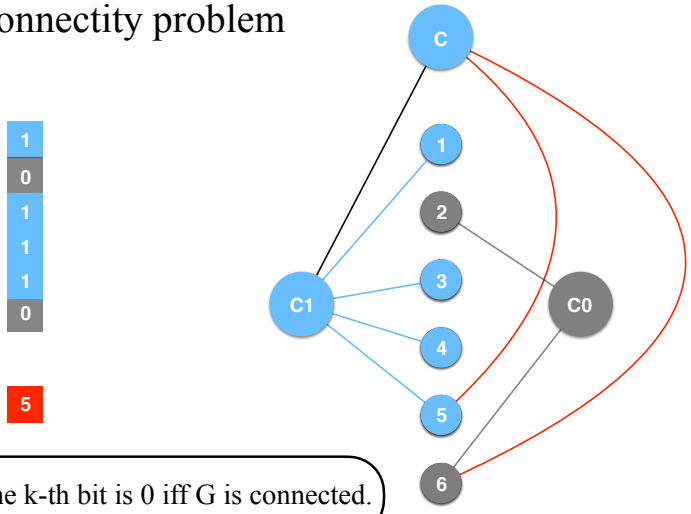
Connectivity

Input: a sequence of edges.

Output: 'Yes' if the graph is connected, or 'No' otherwise.

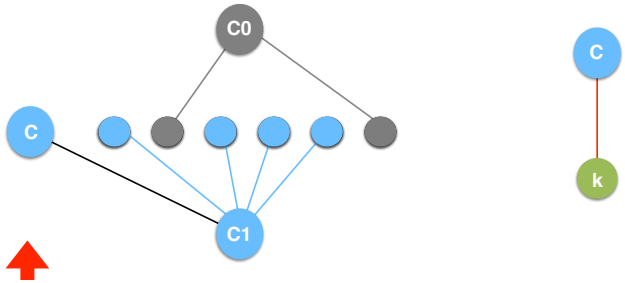
Goal: Prove that any 1-pass streaming algorithm requires $\Omega(n)$ bits.

Reducing the indexing problem to the connectity problem

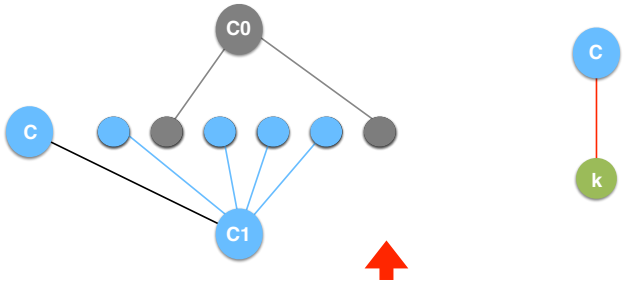


The k-th bit is 0 iff G is connected.

Reducing the indexing problem to the connectity problem

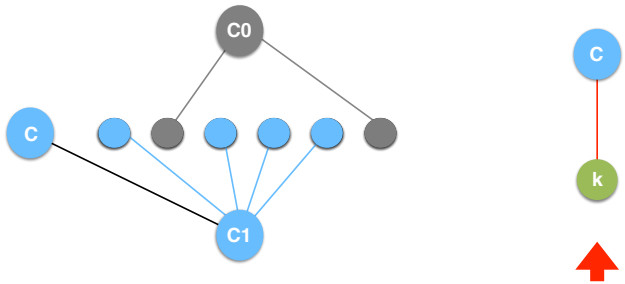


Reducing the indexing problem to the connectity problem



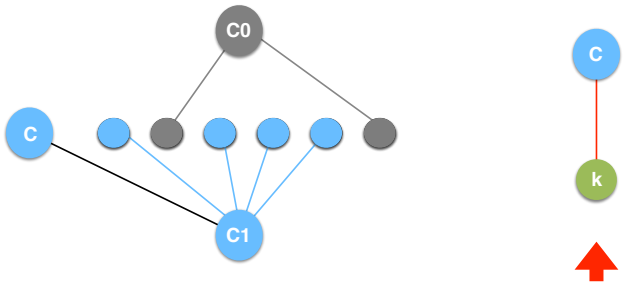
Memory (Message)

Reducing the indexing problem to the connectity problem



Memory (Message)

Reducing the indexing problem to the connectity problem



It thus requires Ω(n)-bit space for the connectivity problem.

Memory (Message)

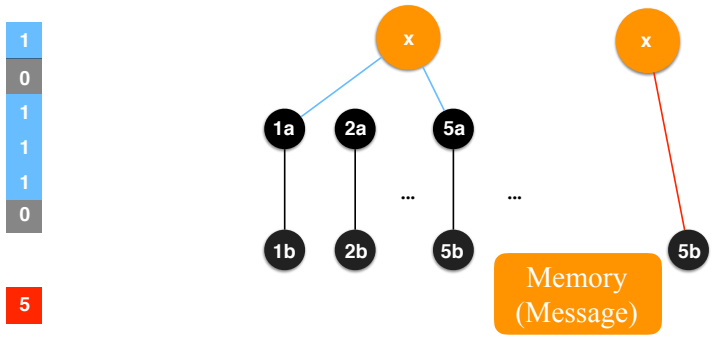
Finding Articulation Points

Input: a sequence of edges.

Output: the set of all articulation points.

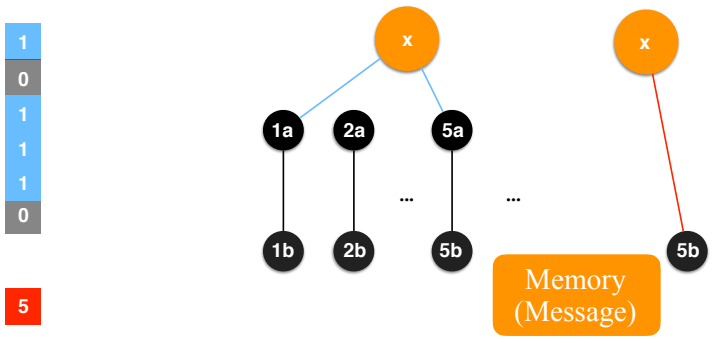
Goal: Prove that any 1-pass streaming algorithm requires $\Omega(n)$ bits.

Reducing the indexing problem to finding articulation points



The k-th bit is 0 iff node kb is an articulation point.

Reducing the indexing problem to finding articulation points



It thus requires $\Omega(n)$-bit space to find articulation points.

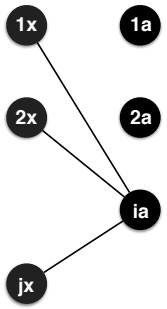
Outputting a BFS Forest

Input: a sequence of edges.

Output: the set of edges in an arbitrary BFS forest.

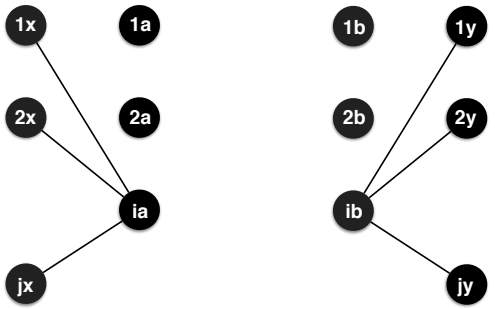
Goal: Prove that any 1-pass streaming algorithm requires $\Omega(n^2)$ bits.

Reducing the indexing problem to outputting a BFS forest



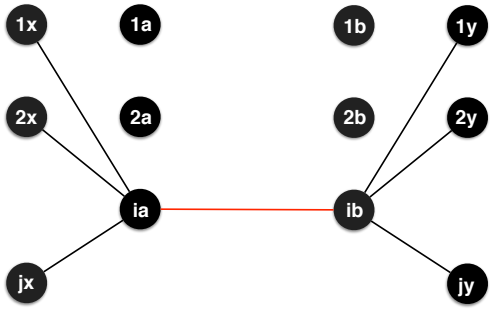
Link an edge (ia, jx) iff (i*n+j)-th bit is 1.

Reducing the indexing problem to outputting a BFS forest



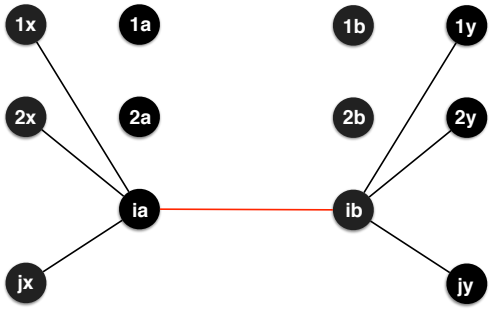
Create a copy of the constructed graph.

Reducing the indexing problem to outputting a BFS forest



Link an edge (ia, ib), and then the (i*n+j)-th bit is 1 iff (ia, jx) or (ib, jy) is an edge in the outputted BFS forest.

Reducing the indexing problem to outputting a BFS forest



It thus requires $\Omega(n^2)$-bit space to output any BFS forest.