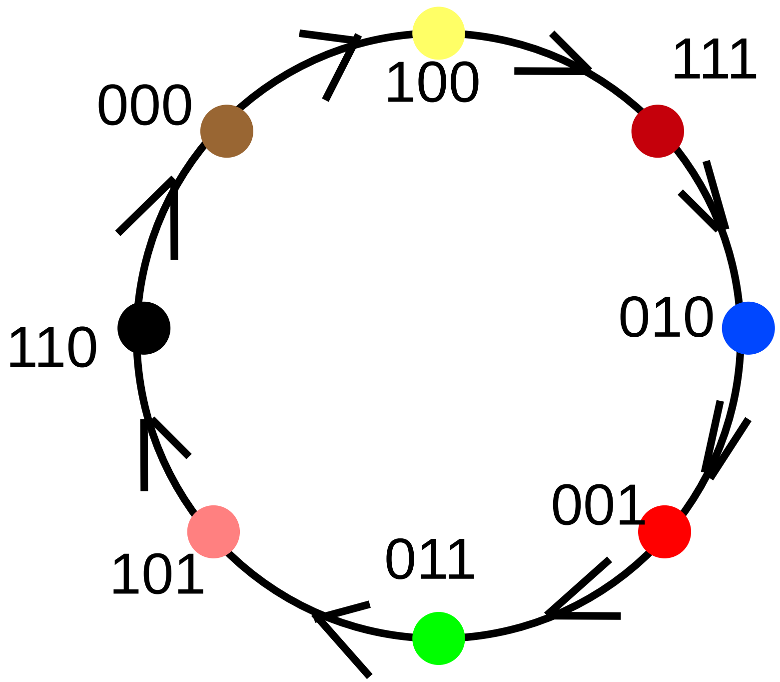
Symmetry Breaking

- A way to induce differences between like (symmetric) participants.
- Useful in applications such as graph coloring
  - Generally, difficult using deterministic techniques.
    - Need randomization
- Special cases where fast, deterministic symmetry breaking can be achieved.
    - Linked lists and directed cycles are an example.

Coloring by Symmetry Breaking

- Consider a directed cycle of n nodes numbered 1 to n.
- Treat the number of the node as its initial color.
- Can reduce colors to log n in one step.
  - Every node u compares its color with that of the successor, and recolors as:
    - $\text{Newcolor}(u) = 2k + \text{color}(u)_k$
  - k is the index of the first bit position that u and v differ from LSB
  - E.g., for 101 and 111, k = 1.



color(u): $0110\boxed{1}01$

i : 2 $color(u)_2 = 1$

u is a node

color(u) : color value in binary

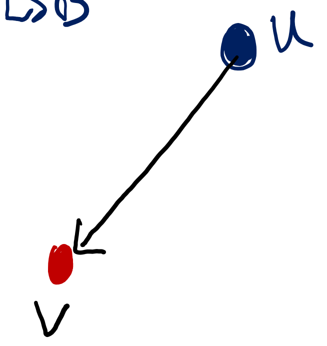
$color(u)_i$: i^{th} bit from LSB in color(u)

$k \leftarrow$ LSB

color(u): $011010\,1$

color(v): $110101\,1$

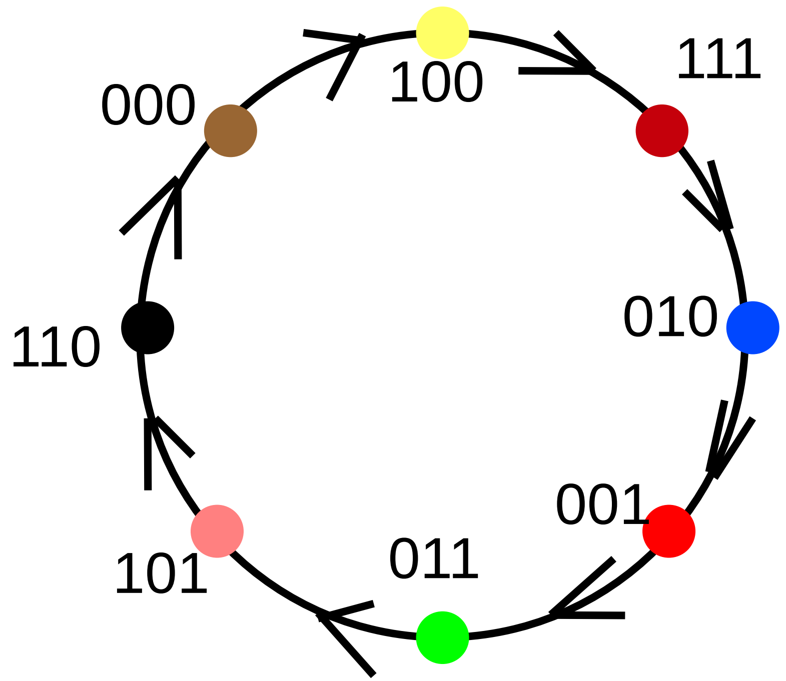
index of LSB = 0



$k = 1$

$color(u)_k = 0$

Coloring by Symmetry Breaking

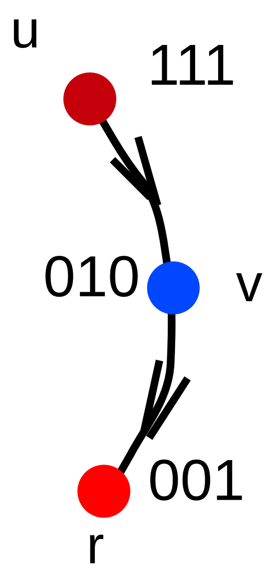


u	v	New Color(u)
110	000	11 (k = 1)
000	100	100 (k = 2)
100	111	00 (k = 0)
010	001	00 (k = 0)
001	011	10 (k = 1)
011	101	11 (k = 1)
111	010	01 (k = 0)
101	110	01 (k = 0)

- Consider a directed cycle of n nodes numbered 1 to n.
- Treat the number of the node as its initial color.
- Can reduce colors to log n in one step.
  - —Compare its color with the successor, Newcolor(u) = 2k + $\text{color}(u)_k$
  - —k is the index of the first bit position that u and v differ from LSB

Coloring by Symmetry Breaking

- Claim: The new colors are valid.
- Proof: Suppose that for u and v such that $u \neq v$, NewColor(u) = NewColor(v).
- Let NewColor(u) = 2k + color(u)$_k$, and NewColor(v) = 2r + color(v)$_r$.
- Let k = r. However, color(u)$_k \neq$ color(v)$_k$. Why?
- Let $k \neq r$. Then, color(u)$_k$ – color(v)$_r$ = 2(r – k).
  - The LHS has an absolute value of at most 1 and the RHS has an absolute value of at least 2.



Coloring by Symmetry Breaking

- In one iteration, can reduce the number of colors from n to 2log n – 1 < 2 log n.
  - Initial colors are log n bits
  - New colors are only 1+ [loglog n] bits.
- Can we repeat again?
  - Yes.
  - Reduces number of bits from t to 1+[log t].
  - But, at some point t < 1 + [log t]. No advantage any further.
  - Happens at t = 3.
- So, repeat till only 8 colors are used.

Coloring by Symmetry Breaking

- At that point, can still reduce the number of colors as follows:
- For i = 8 downto 3 in sequence
  - If node u is colored i, then u chooses a color among {1,2,3} that is not same as the colors of its neighbors.
- Possible to do so. Why?

Coloring by Symmetry Breaking

- At that point, can still reduce the number of colors as follows:
- For i = 8 downto 3 in sequence
  - If node u is colored i, then u chooses a color among {1,2,3} that is not same as the colors of its neighbors.
- Possible to do so. Why?
  - Each node has only two neighbors.
  - So, only some two colors amongst {1,2,3} can be used up already.

Coloring by Symmetry Breaking

- Total time analyzed as follows:
  - Each iteration of symmetry breaking reduces number of bits from to $1+\lceil \log t \rceil$.
  - The recurrence relation is $T(n) = T(\log n) + 1$
  - Solution: $T(n) = O(\log^* n)$.
  - In the next phase, only 5 iterations.
  - So, overall time = $O(\log^* n)$
- Work however is $O(n\log^* n)$.
- $\log^* n = i$ such that

$$\underbrace{\log(\log(\ldots\ldots(\log}_{i} n))) = 1;$$

- The algorithm extends to lists and rooted trees also.

Coloring to Independent Sets

- For bounded degree graphs colored with O(1) colors, a coloring is equivalent to finding a large independent set.

- Iterate on each color and count the number of nodes with a given color.
- Pick the subset of like colored nodes of the largest size.
  - Clearly, an independent set.
  - Has a size of at least a fraction of n.