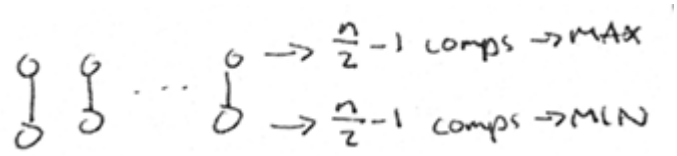
Adversary Approach

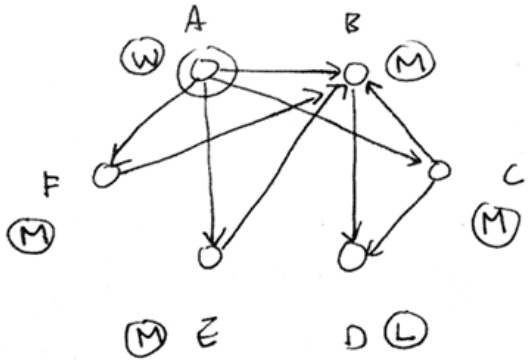
April 2, 2014 15:06

The **adversary** approach to problem complexity lower bounds

- *Ex:* Problem P: "Find MIN and MAX of set S of n elements"
  - (Assume n is even)
  - The naïve scan S twice
    - $n-1$ comps to find MAX
    - $n-2$ comps to find MIN
    - Total of $2n-3$ comps
  - Better algorithm:
    - $n/2$ pairs:
      - 

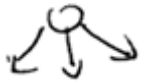
    - Total: $\frac{3n}{2}-2$ comps

- **Theorem**: any comparison-based algorithm for solving P requires at least $\frac{3n}{2}-2$ comps in the worst-case

- 

  - 9 arrows

- 4 types of nodes:

		Initially	At the end
N	Never compared	n	0
W	 Always won so far	0	1
L	 Always lost	0	1
M	 Mixed (won some, lost some)	0	$n-2$

- Adversary: "delay the creation of 'M's"
- All pairs possible:
N, N	W, L
N, W	L, W
N, L	W, L
N, M	*Doesn't matter* W, M
W, W	*Both can't win, one will become mixed* **W, M**
W, L	W, L
W, M	W, M
L, L	*One will win, becomes mixed* **L, M**
L, M	L, M
M, M	M, M

  - Note: M's only created from W or L
- Starting from n elements of type "N", algorithm must create:
  1. $(n-2)$ "M"s
     - To create each "M", needs to do 1 comparison of type (*) [transforms a "W" or an "L" into an "M"]
     - ⇒ Algorithm must do at least $n-2$ comparisons of type (*)
  2. $(n-2)$ "W" or "L"s that later become "M"s
     - 1 W that remains
     - 1 L that remains
     - Total: n "W" or "L"
     - Algorithm needs to do at least $\frac{n}{2}$ comparisons to create them!
  - So algorithm must do at least $(n-2)+\frac{n}{2}=\frac{3n}{2}-2$ comparisons