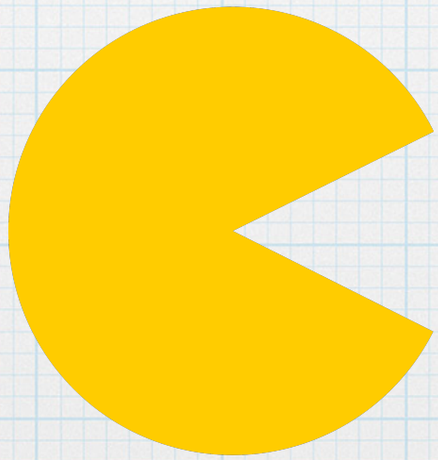
CS 188 Discussion 1: More Search, plus Heuristics!

TA: Sherdil Niyaz



Reminders

* Project 1 is out. Please start! Covers search.
* Office hours exist to help you- use them!
* **HW1 Deadline extended to today at midnight due to typo in lecture slides!**
* Feedback available on section site:

  http://sniyaz.weebly.com/cs188.html

The Search Combo Meal

Fringe Strategy?

Stack -> DFS

Queue -> BFS

PQ with g(n) -> UCS

PQ with g(n) + h(n)-> A*

+

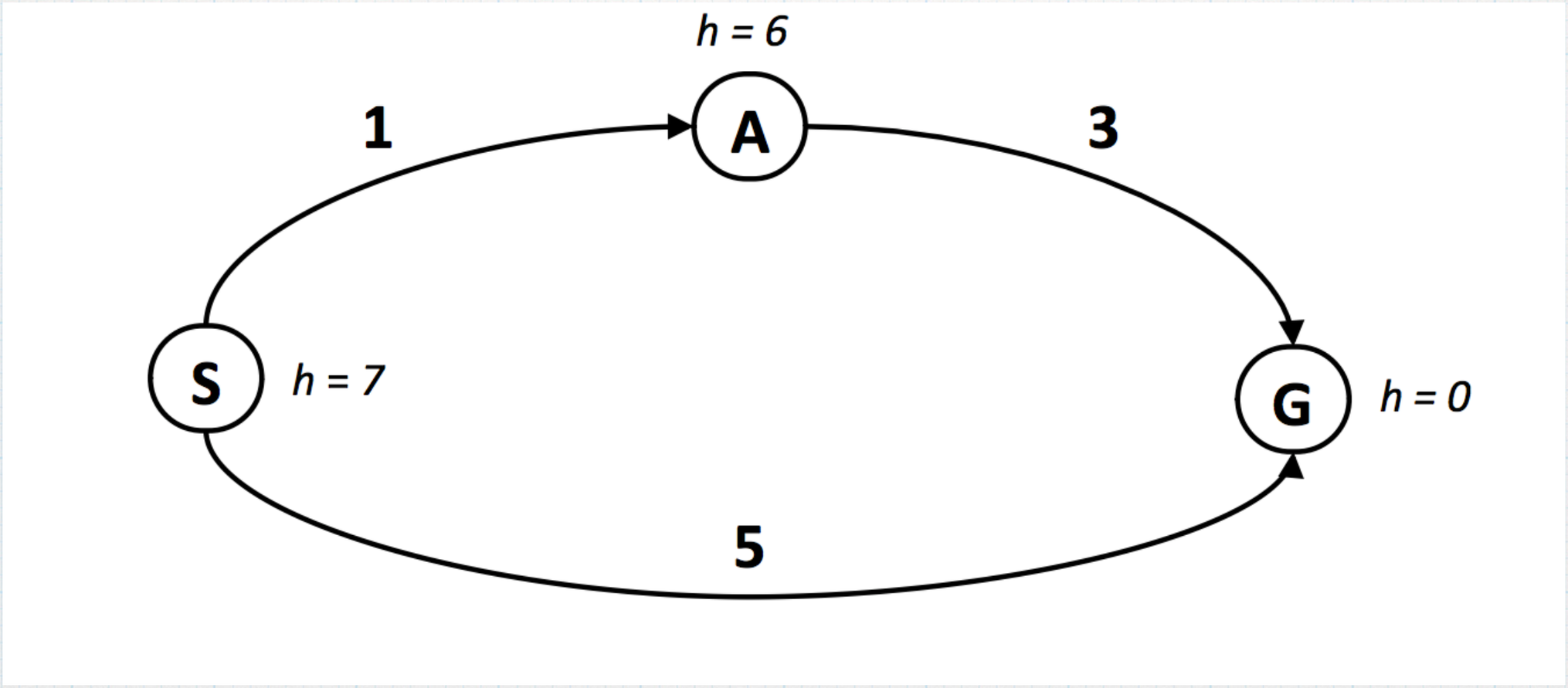
Use a Closed Set?

Yes -> Graph Search

No -> Tree Search

Hueristics

* One way to think about it: a "guess" of how far a certain state is from the goal.

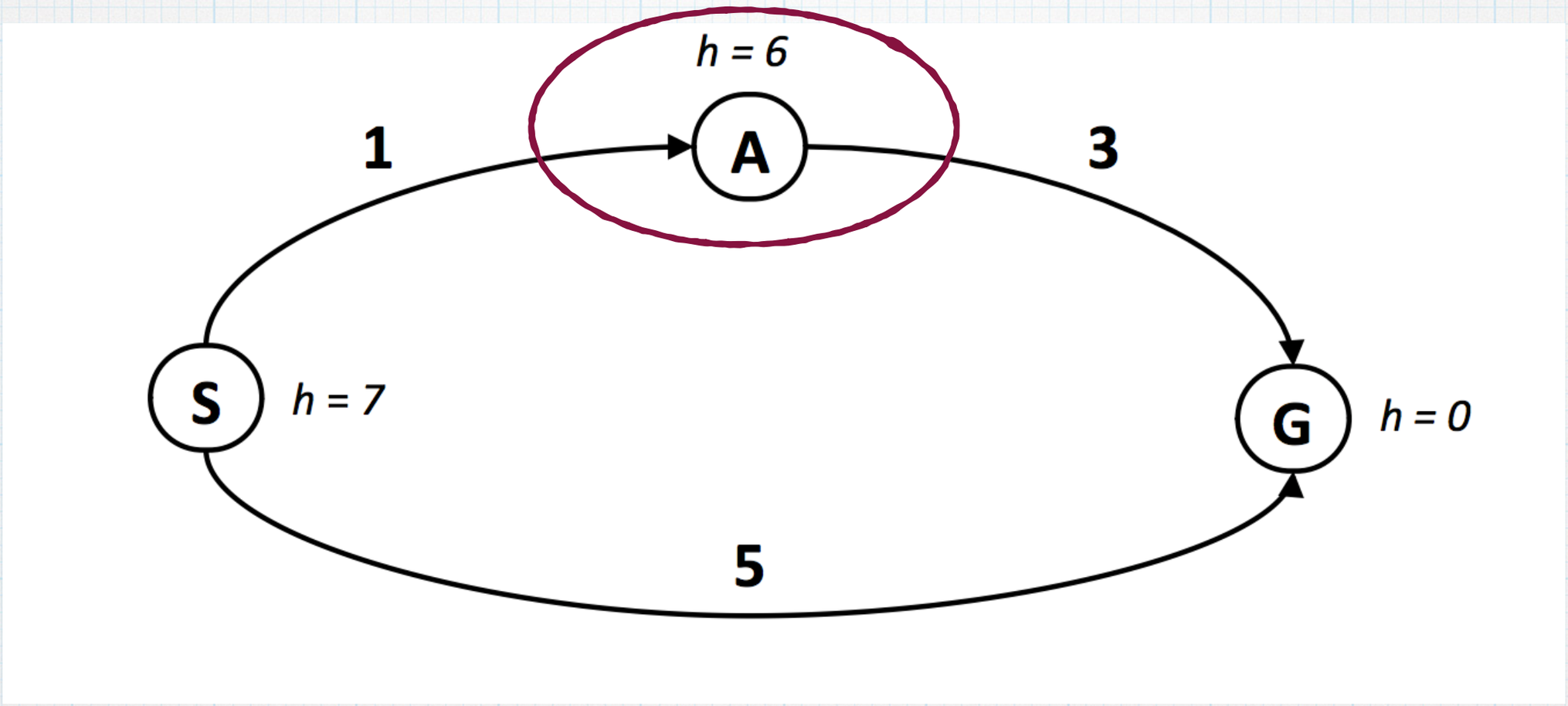


Hueristics: Admissibility

* The "guess" at each node must be **less** than the **real** cost to the goal from that node!

* (Assuming that we take the optimal path to the goal)

Hueristics: Admissibility

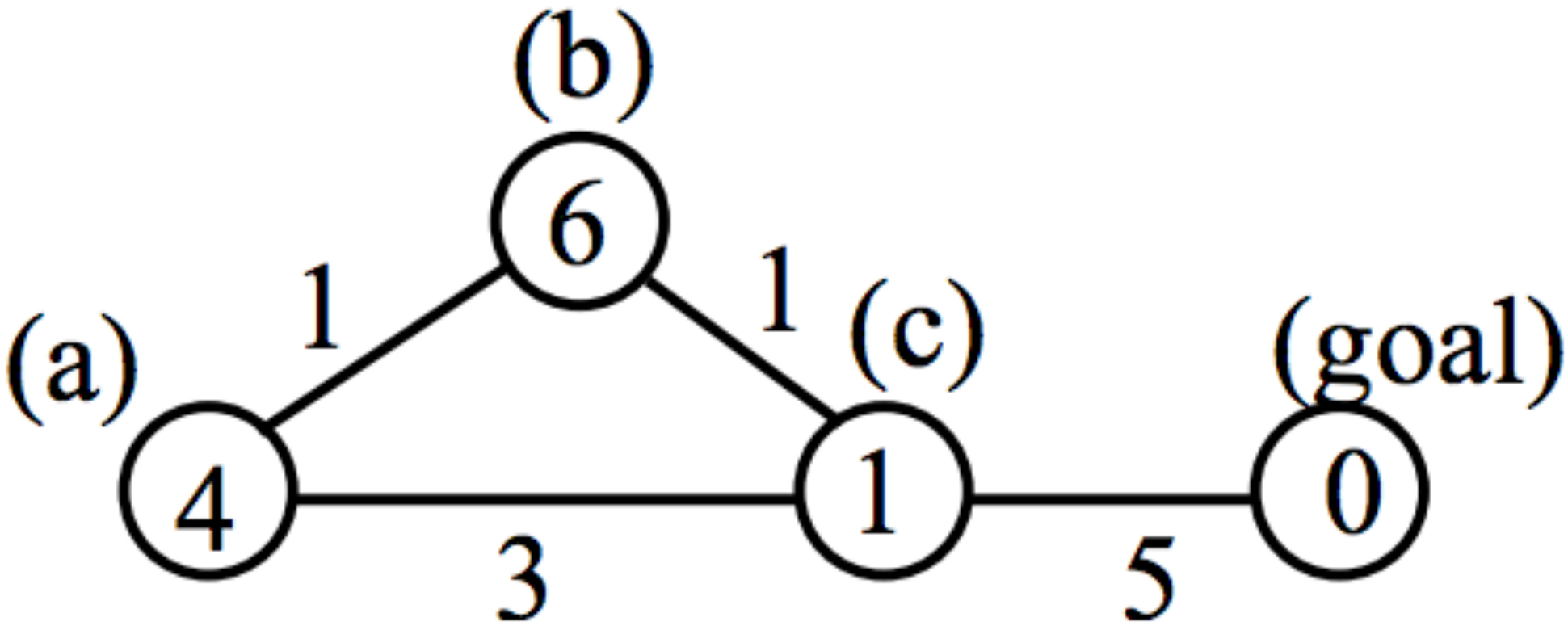


* Whoops.
* Theorem: Without an admissible heuristic, A* **tree** search is not optimal.

Hueristics: Consistency

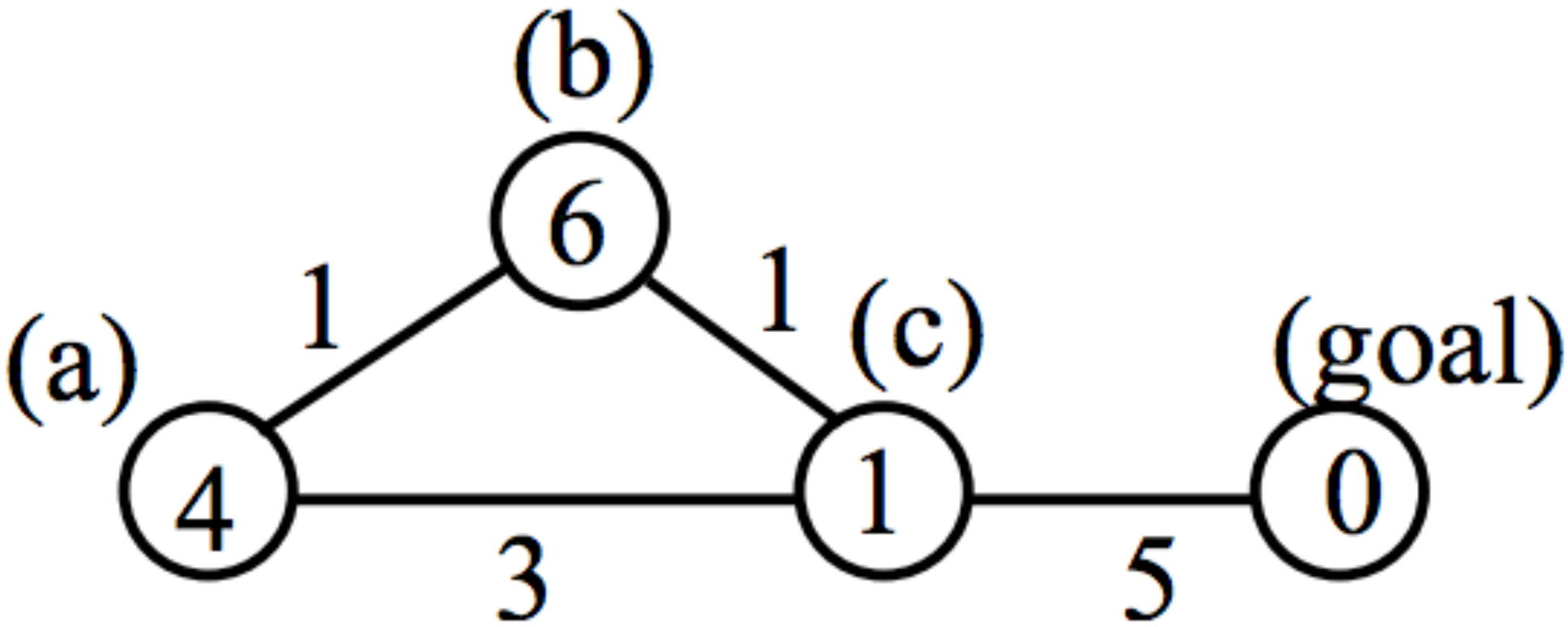
* A **stronger** statement than admissibility!
* The heuristic "drop" between nodes should be **less** than the cost of the edge connecting them.
* No longer a condition only involving every node relative to the goal.
* Now a condition between individual nodes!

Admissible?



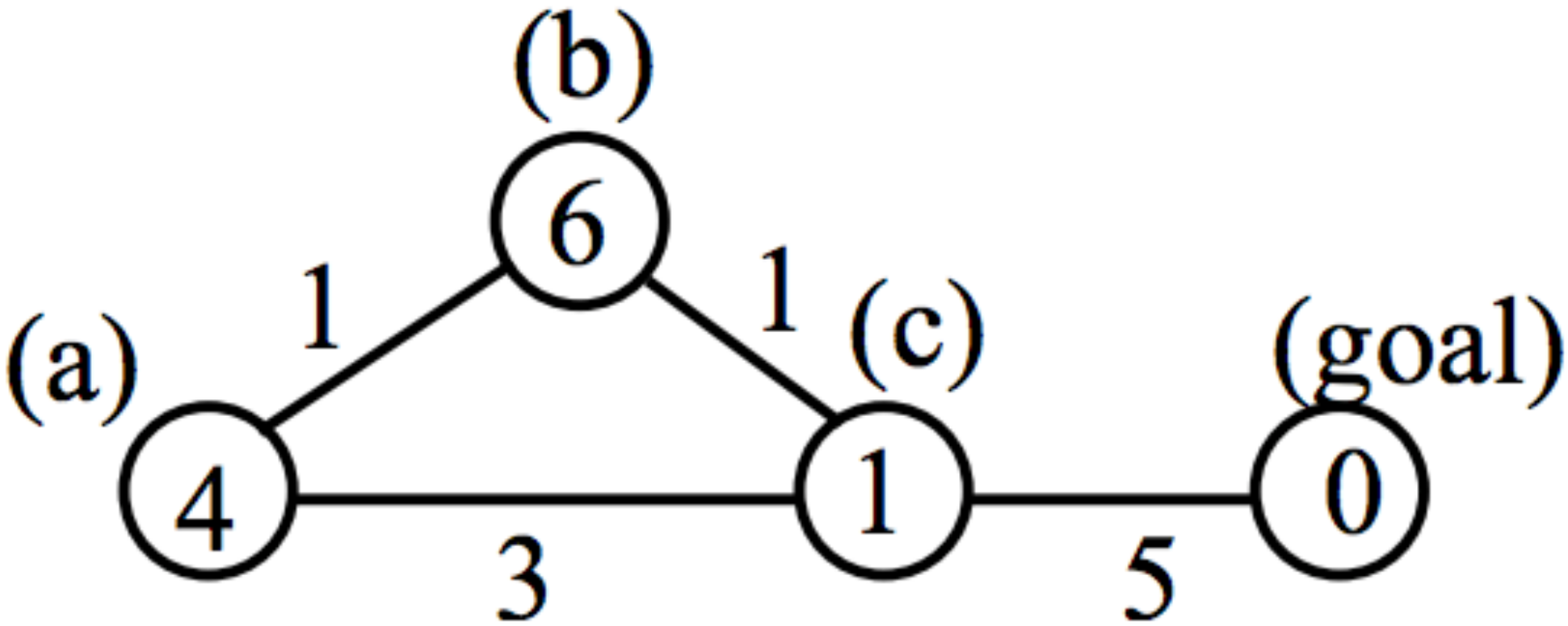
* (h values written inside each node)

Admissible?

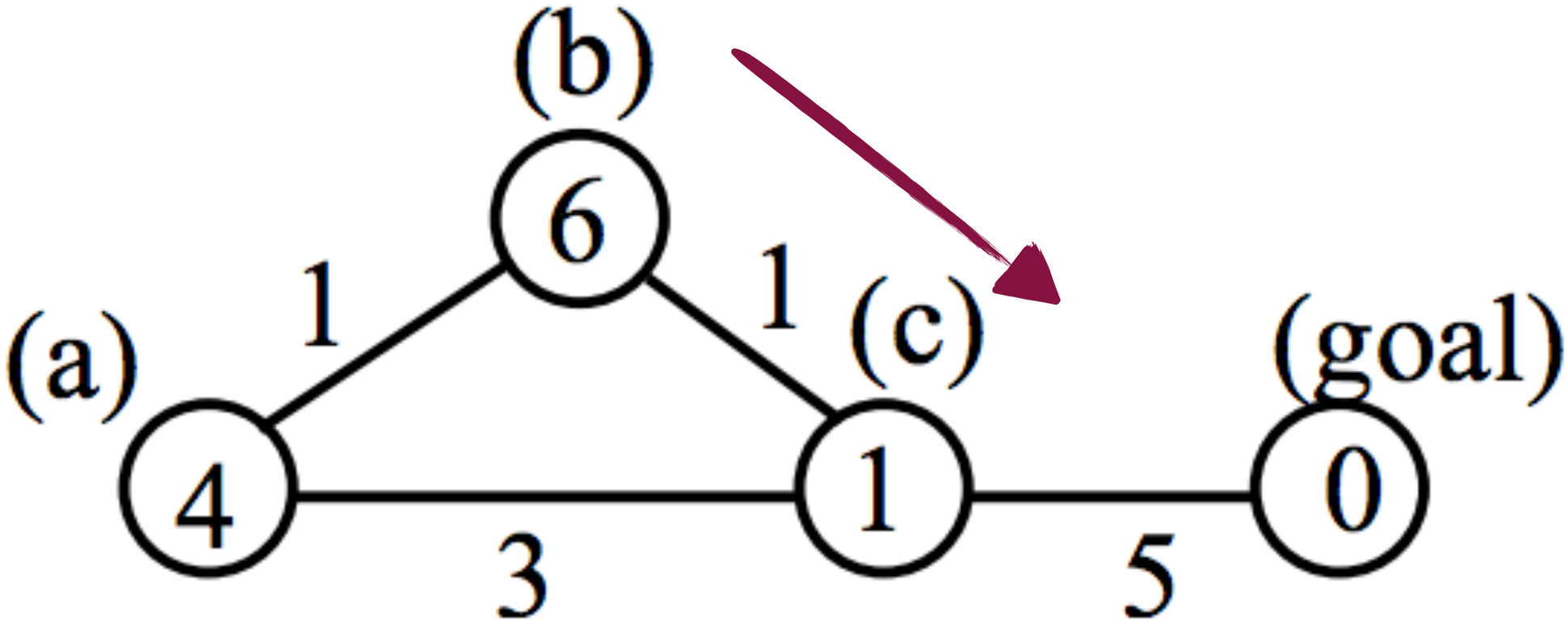


* Yes! :)

Consistent?



Consistent?



* No!

Hueristics: Consistency

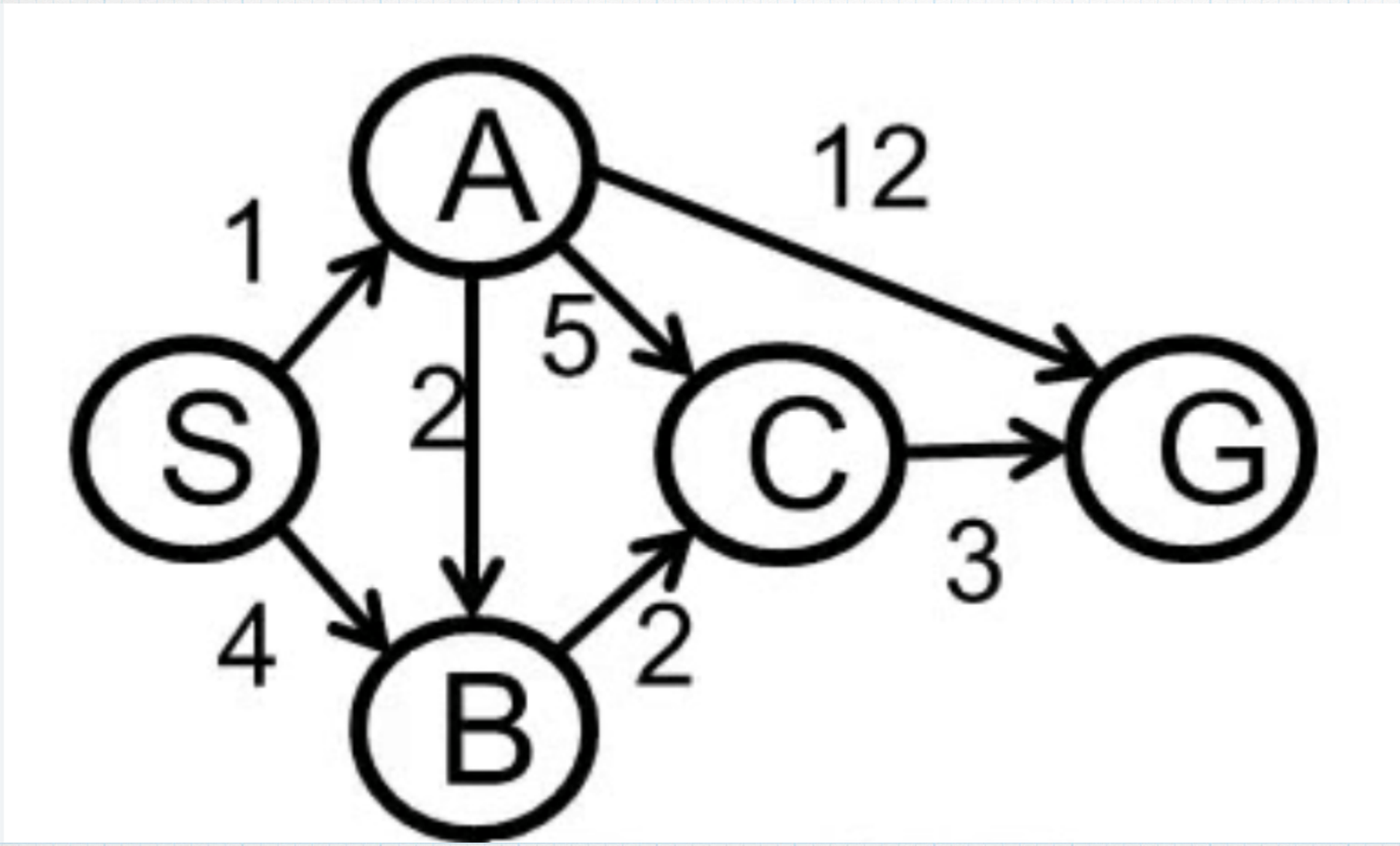
* Theorem: Without consistency, A* **graph** search is non optimal!

* The reason: if we have a heuristic that isn't consistent, the cost of a path (g(n) + h(n)) might **drop** as we traverse it.

* **For hotshots:** Why does this break A* graph search but not tree search?

* The answer is complicated...but worth thinking about.

A* Example



State	H
S	7
A	6
B	2
C	1
G	0