- A* is not optimal because we need estimated costs $h(n)$ to be less than actual costs

ADMISIIBLE HEURISTICS

- A heuristic h is **admissible** (optimistic) if $0 \leq h(n) \leq h^*(n)$, where $h^*(n)$ is the true cost to a nearest goal.
- Designing admissible heuristics is most of what's involved in using A* in practice

Optimality of A* Tree Search Proof in slides

CREATING ADMISSIBLE HEURISTICS

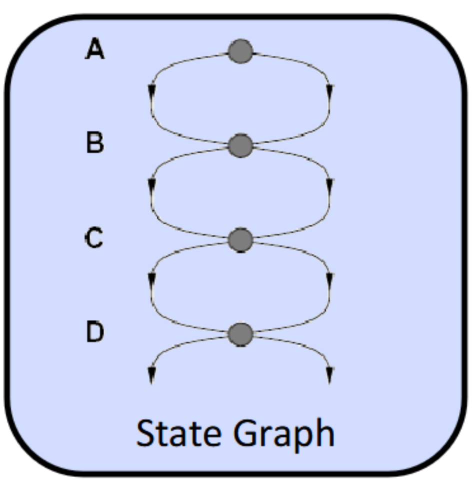
- Most work in solving search problems optimally is in designing admissible heuristics
- often, admissible heuristics are solution to relaxed problems where new actions are available
- Inadmissible heuristics are often useful too

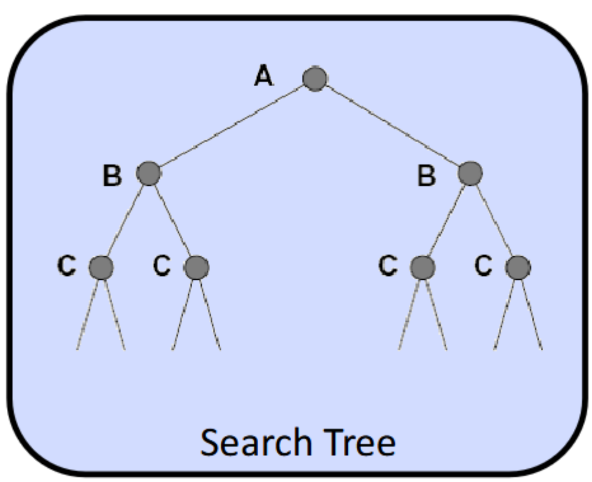
For A*:

- trade-off between quality of estimate and work per node
- heuristics get closer to true cost, we will expand fewer nodes but usually do more work per node to compute the heuristic itself
- max of admissible heuristics is admissible, and it is a "better" heuristic

GRAPH SEARCH

Why?
In tree search, there is failure to detect repeated states can cause exponentially more work



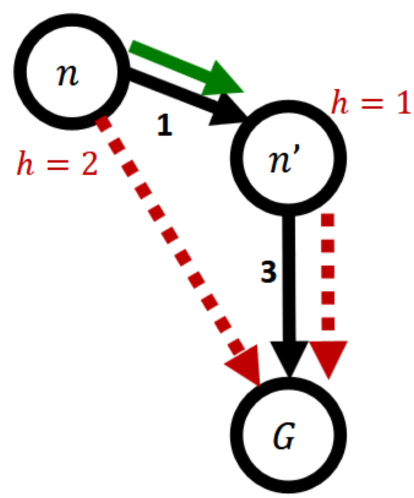


Idea:

- Never expand a state twice
  How to implement:
- Tree search + set of expanded states ("closed set")
- expand the search tree node by node, but....
- before expanding a node, check to make sure its state has never been expanded before
- if not new, skip it, if new add to closed set
- store the closed set as a set, not a list (IMPORTANT)

```
function GraphSearch(problem) <- return a solution or failure
frontier <- {Initial state}; explored <- null //Graph search + Uniform
repeat                                              // Cost = Dijkstra's Algorithm
    if frontier = null then return failure
    node <- POP(frontier)
    if IsGoal(node) then return the corresponding solution
    if node not in explored:
        explored <- explored U {node}
        for a in Actions(node):
            a' = Result(node, a)
            INSERT(frontier,a')
```

- **Main idea:** estimated heuristic costs ≤ actual costs
  - Admissibility: heuristic cost ≤ actual cost to goal
    - $h(n)$ ≤ **actual cost from n to G**
  - Consistency: heuristic "arc" cost ≤ actual cost for each arc
    - $h(n) - h(n') \leq cost(n\ to\ n')$
    - $h(n) \leq cost(n\ to\ n') + h(n')$



- **Formally:** A heuristic $h(n)$ is consistent, if for every node n and every successor n' of n
  - $h(n) \leq \text{cost}(n \text{ to } n') + h(n')$

OPTIMALITY:

TREE SEARCH

- A* is optimal if heuristic is admissible
- UCS is a special case ($h(n) = 0$ for all n)

GRAPH SEARCH

- A* is optimal if heuristic is consistent
- UCS optimal ($h = 0$ is consistent)

Note: Consistency implies admissibility

- In general, most natural admissible heuristics tend to be consistent, especially if from relaxed problems.