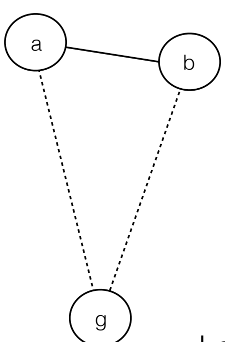
Single Agent Se

Lecture 4
Heuristics, A*, IDA*

UNIVERSITY of DENVER
DANIEL FELIX RITCHIE SCHOOL OF ENGINEERING & COMPUTER SCIENCE

Final Heuristic Details

Consistency (Triangle Inequality)

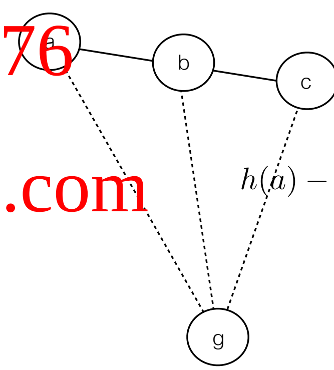


$$h(a) \le c(a, b) + h(b)$$

$$h(a) - h(b) \le c(a, b)$$

Local, since a & b are neighbors

Consistency (Triangle Inequality)



$$h(a) - h(b) \le c(a, b)$$

$$h(b) - h(c) \le c(b, c)$$

$$h(a) - h(b) + h(b) - h(c) \le c(a, b) + c(b, c)$$

$$h(a) - h(c) \le c(a, c)$$

This step requires c(a, c)= c(a, b)+c(b, c)

Global, since a & c are not neighbors

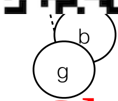
UNIVERSITY of DENVER
DANIEL FELIX RITCHIE SCHOOL OF ENGINEERING & COMPUTER SCIENCE

Admissibility from Consistency

- What if b and g are the same node?
- Local consistency:

$$h(a) \leq h(b) + c(a,b)$$
$$h(a) \leq 0 + c(a,b)$$

- Global consistency is the same
- Consistent heuristics must be admissible



UNIVERSITY of DENVER
DANIEL FELIX RITCHIE SCHOOL OF ENGINEERING & COMPUTER SCIENCE

Pure Heuristic Search

- Best-First Algorithm
- f = h()
- Complete? / Will it find a solution?
  - Only on finite graph
- Optimal?
  - No
- Space, Time?
  - Undetermined

A*

UNIVERSITY of DENVER
DANIEL FELIX RITCHIE SCHOOL OF ENGINEERING & COMPUTER SCIENCE

A*

- Best-First Search
  - f = g+h
  - f is an estimate of the complete path length
- Optimality?
  - Not optimal unless we guarantee properties of our heuristic

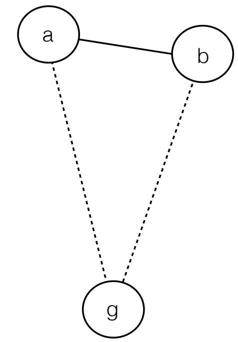
A* Optimality

- Property #1
  - f-costs along any optimal path are [illegible] increasing
    - Assume non-negative costs
    - Assume consistent heuristic

A* Consistency

- f-cost is monotonic non-decreasing

$$h(a) \le h(b) + c(a,b) \quad \text{(consistency)}$$
$$g(a) + h(a) \le g(a) + h(b) + c(a,b)$$
$$f(a) \le g(b) + h(b)$$
$$f(a) \le f(b)$$



Or, we won't follow the path from *a* to *b* in the search.

A* Optimality

- Property #1(a)
  - The f-cost of the start (and all states) is non-overestimating
    - Follows from admissibility/consistency

A* Optimality

- Property #2
  - A node on the optimal path (to any reachable state) is always on the open list with its optimal g-cost from the start. (Proof by induction)
    - Initially start is on OPEN
    - Assume step n; step n+1:
      - If node at n+1 is on optimal path, its successor will be on open
      - If not, the previous node will still be on OPEN

A* Optimality

- Property #3
  - Every closed node has optimal g-cost
  - Initially no states on closed
  - Assume at step n all closed states hav
    - What if the state expanded at step r optimal g-cost?
      - There would be a node with lowe
      - We would expand that node next
      - Therefore the node at n didn't have minimal g-cost

A* Optimality

- Property #4
  - Every node with finite cost will eventually be expanded (assume min edge cost w.l.o.g.=1)
  - For any given depth *d*, there are at most N(b, d) nodes at depth <= d.
  - This is finite, so for any finite cost only a finite number of steps are needed.

$$N(b,d) = \frac{b^{d+1}-1}{(b-1)}$$

A* Optimality

- What can happen:
  - Run out of nodes on open
    - Can't happen if path exists (property #2)
  - Expand nodes forever and never reach goal
    - Assume minimum edge cost
    - Assume bounded branching factor
  - Expand the goal
    - Found it with optimal cost (property #3)
    - Find in finite # of steps (property #4)

A* Optimality Proof (1)

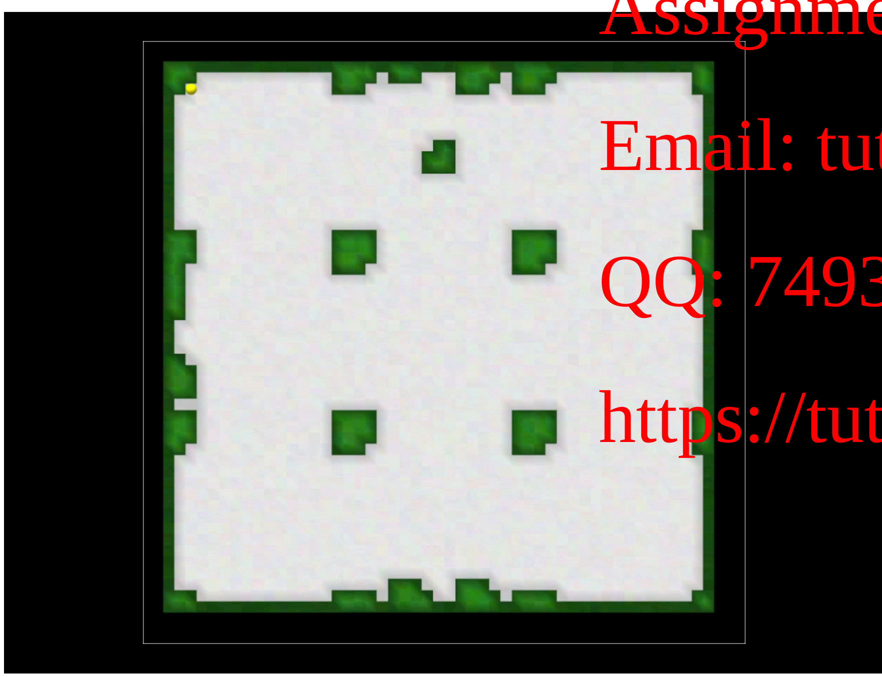
- Assume there exists an algorithm B that expands less nodes that A* on problem P
  - There must be some node *m* not expanded by B, but expanded by A*
  - f(m) = g(m) + h(m) < c
  - Create a new problem P' with new goal g'
  - Add an edge with cost h(*m*) to g'

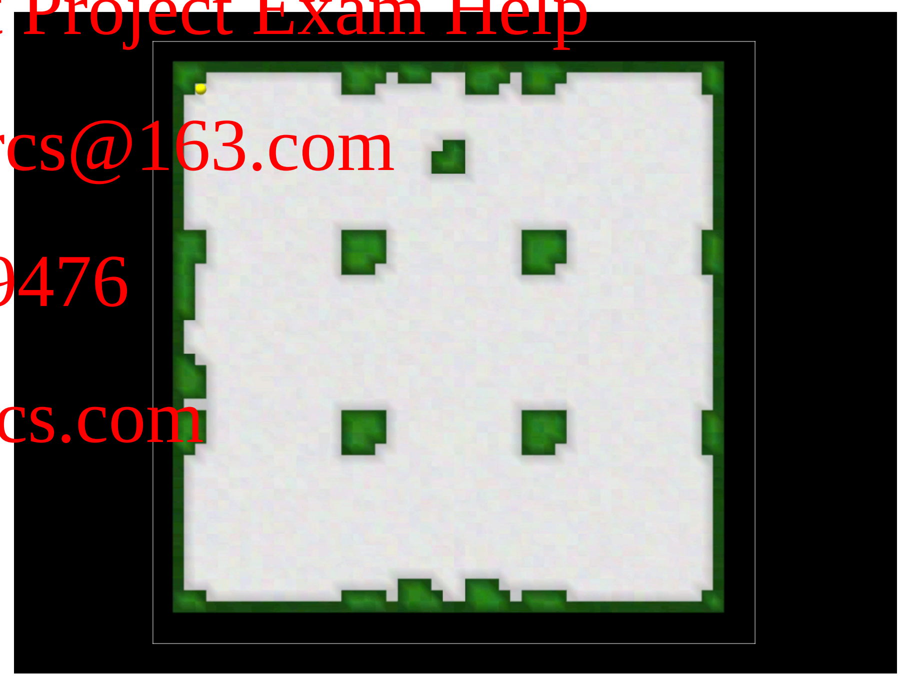
A* Optimality Proof (2)

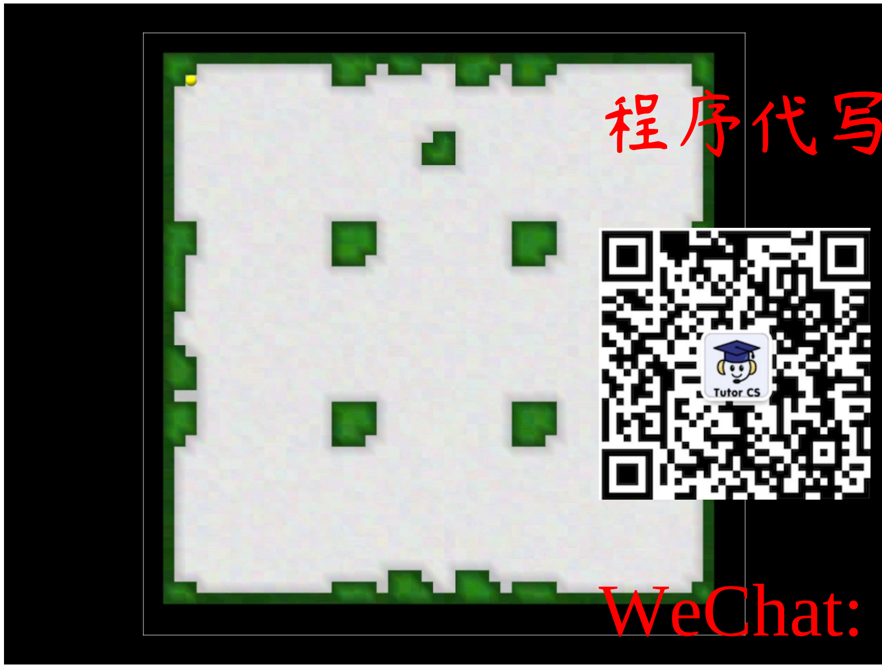
- Goal g' is now shorter than the original goal
- g(g') = g(m) + c(m, g') = g(m) + h(m
- g' will not be found by B, but will be
- Is our heuristic still consistent/admi
  - |h(m)-h(g')| ≤ c(m, g')
  - |c(m, g') - 0| ≤ c(m, g')

Can we make A* go faster?

- A*: f(n) = g(n) + h(n)
- Weighted A*: F(n) = (1-*w*)·g(*n*) + *w*·h(*n*)
- If *w* = 1?
  - Pure Heuristic Search
- If *w* = 0?
  - Dijkstra's







IDA*

IDA*

- Previously, BFS → DFS → DFID
- Now, A* → IDA*
- Perform DFS within f-cost limits
- Korf, 1985

IDA* Pseudo-Code

```
IDA*(start, goal)
limit ← f-cost(start)
do
   path = cost-limited-DFS(start, goal, limit)
   limit ← newlimit
while (!path)
return path
```

Cost limits

- How do we determine the next cost limit?
  - Keep track of the minimum f-cost limit found during search
  - This is the next limit