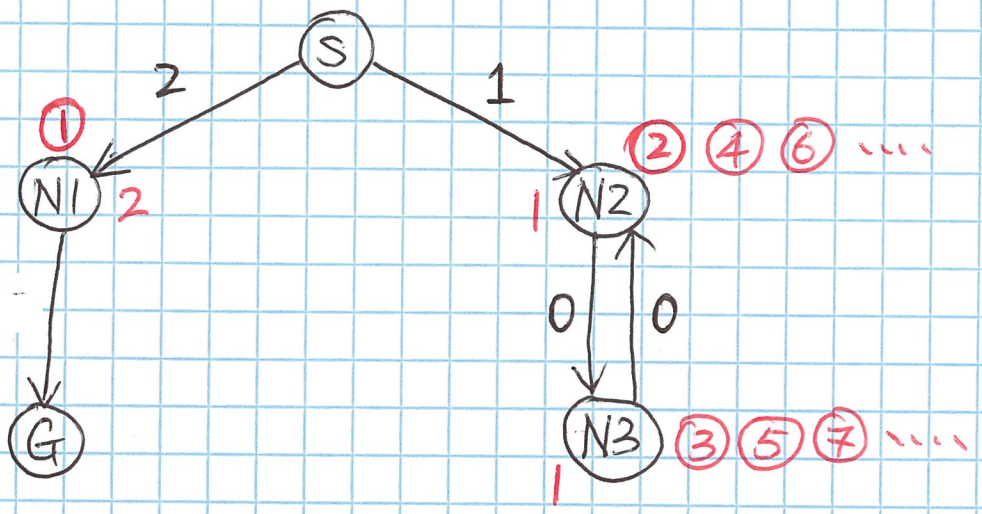
Lowest-Cost-First Search

Lecture 4.

Not complete in general.

Complete if every arc cost exceeds $\varepsilon > 0$ and b is finite.



Example of ~~zero~~-cost action:

NoOp in assembly language.

Time and Space Complexity:

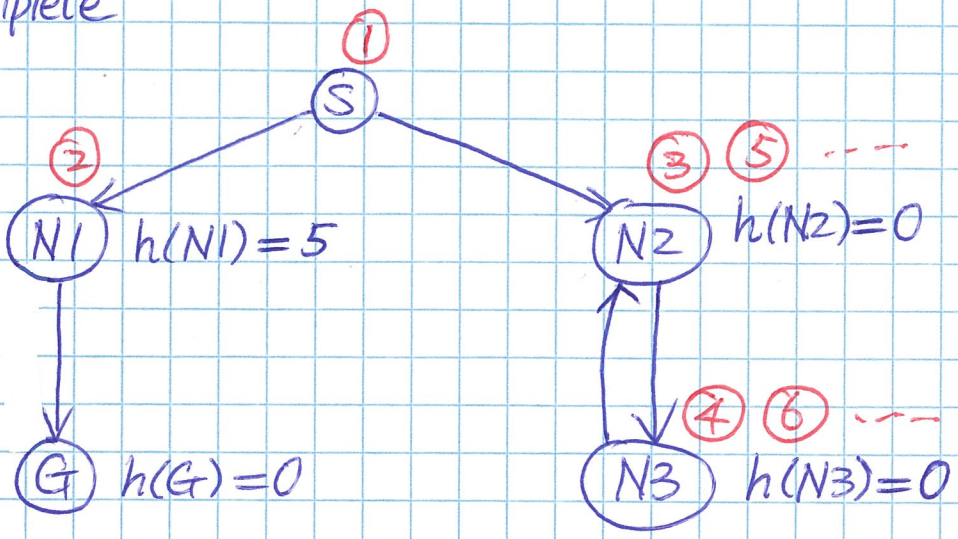
In the worst case, we'll reach level $1 + C^*/\varepsilon$.

- C^* is cost of optimal path.
- Every step has cost $\geq \varepsilon$
- Takes at most C^*/ε to find optimal goal.
- "1+" we test a node when it's expanded, not when it's generated.

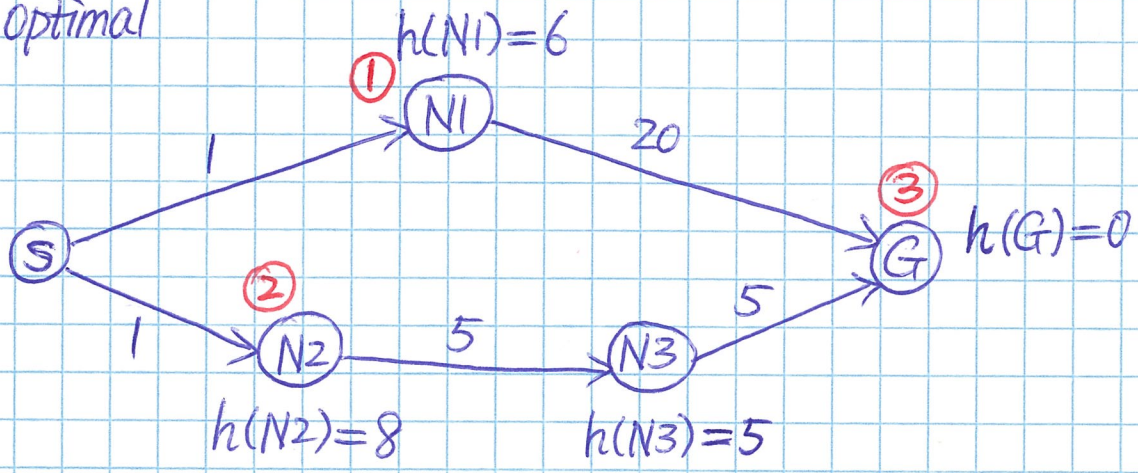
Greedy Search.

Lecture 4.

Not complete



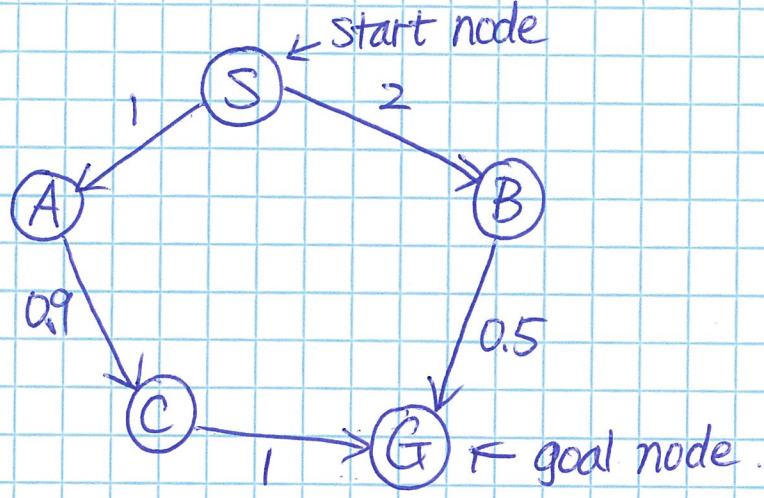
Not optimal



Lowest-Cost-First Search (Lecture 4)

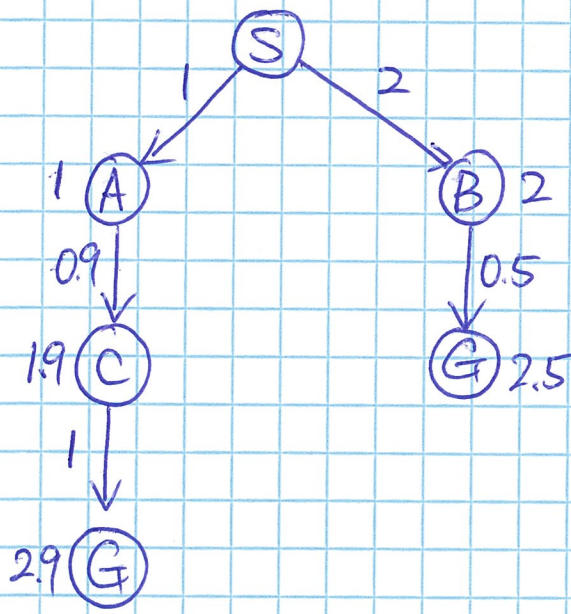
- LCFS = Dijkstra's shortest path algorithm.

search graph



search tree.

A node is circled when it's expanded.



frontier : (S^0)

frontier : (A^1), B^2

frontier : B^2, $(C^{1.9})$

frontier : (B^2), $G^{2.9}$

frontier : $G^{2.9}$, $(G^{2.5})$

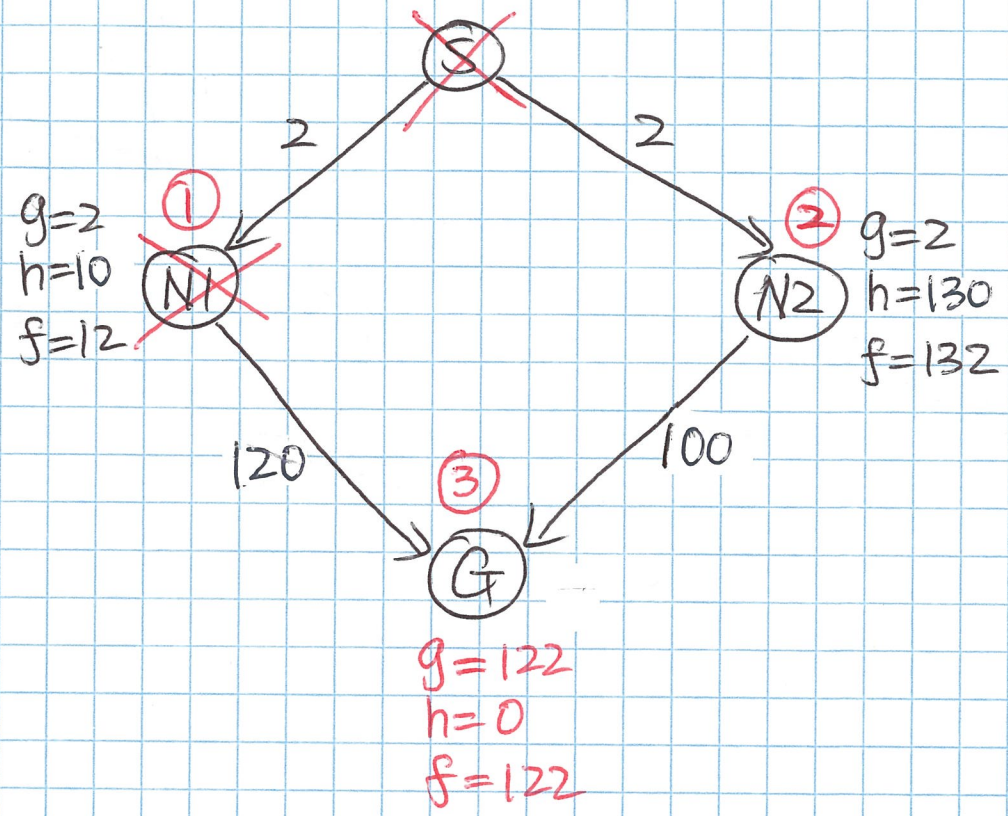
return path $S \rightarrow B \rightarrow G$.

Make sure you understand :

- what's a state space?
- difference between search graph & search tree.
- difference between generating a node and expanding a node.
- we test for goal when a node is chosen for expansion.

A* Search

A* is not optimal if h is NOT admissible.



stop and return path S → N1 → G

which is NOT optimal.