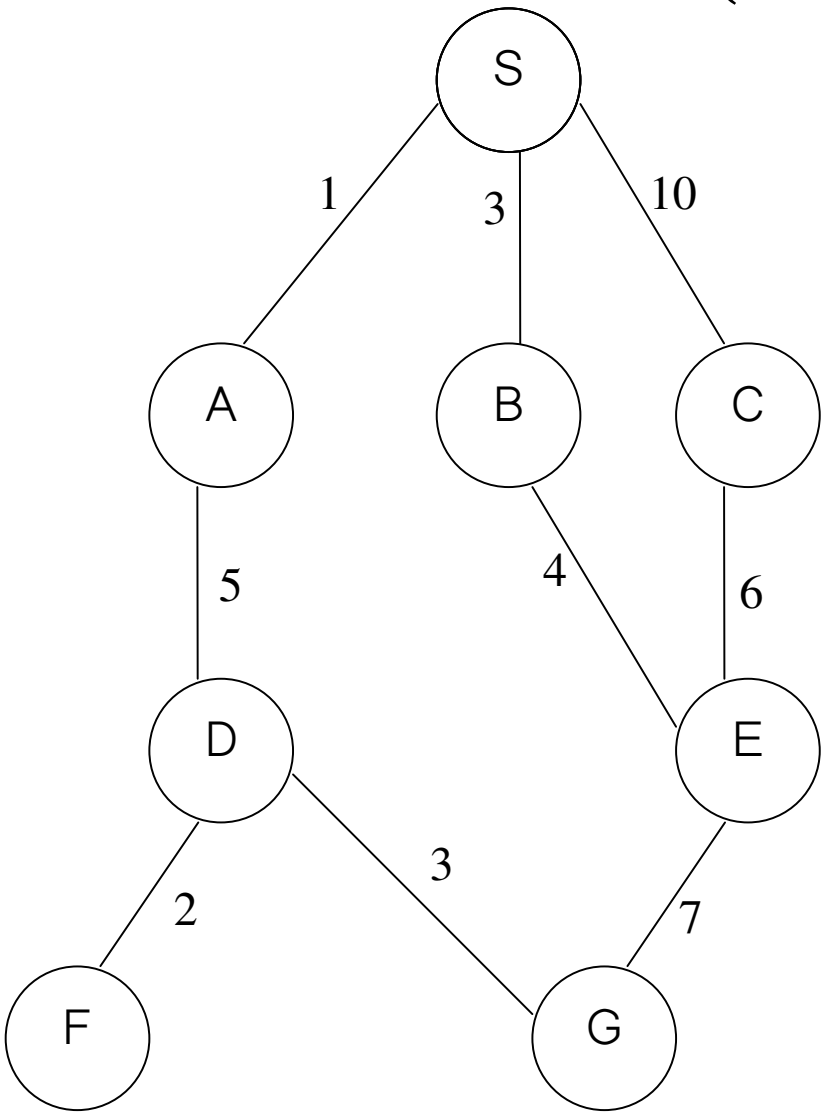
CS 344

Artificial Intelligence

By Prof: Pushpak Bhattacharya

Class on 15/Jan/2007

General Graph search Algorithm (Review)



Graph G = (V,E)

1) Open List : S $^{(Ø, 0)}$

Closed list : Ø

2) OL : $A^{(S,1)}$, $B^{(S,3)}$, $C^{(S,10)}$

CL : S

3) OL : $B^{(S,3)}$, $C^{(S,10)}$, $D^{(A,6)}$

CL : S, A

4) OL : $C^{(S,10)}$, $D^{(A,6)}$, $E^{(B,7)}$

CL: S, A, B

5) OL : $D^{(A,6)}$, $E^{(B,7)}$

CL : S, A, B , C

6) OL : $E^{(B,7)}$, $F^{(D,8)}$, $G^{(D, 9)}$

CL : S, A, B, C, D

7) OL : $F^{(D,8)}$, $G^{(D,9)}$

CL : S, A, B, C, D, E

8) OL : $G^{(D,9)}$

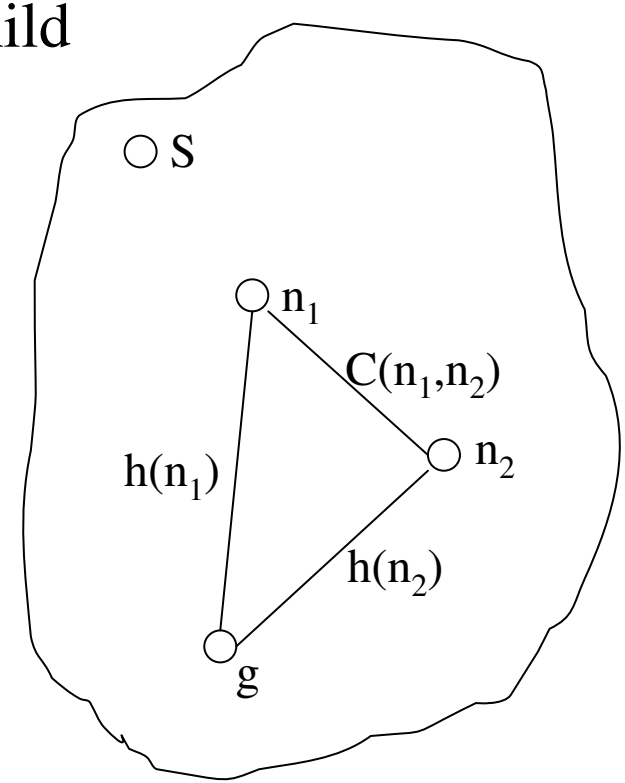
CL : S, A, B, C, D, E, F

9) OL : Ø

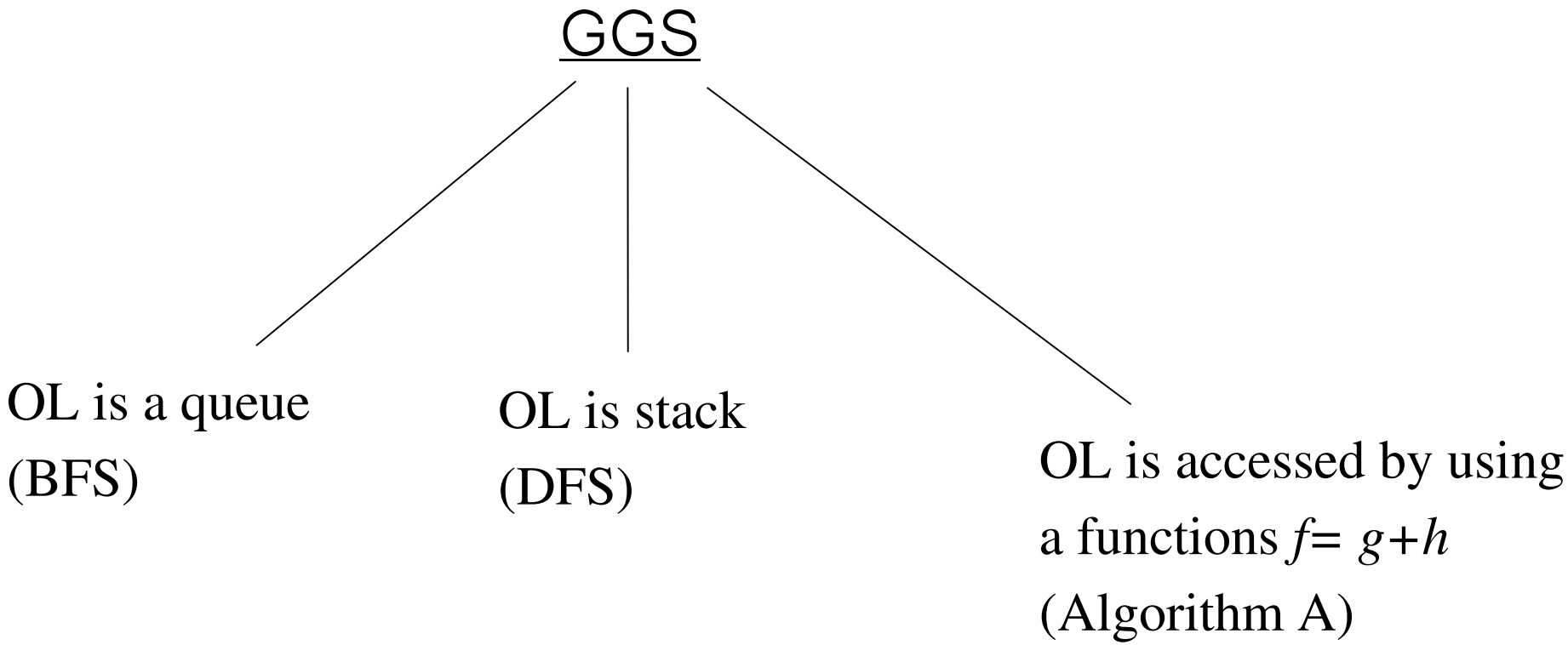
CL : S, A, B, C, D, E, F, G

GGS Review (contd.)

- Key data structures : Open List, Closed list
- Nodes from open list are taken in **some order**, expanded and children are put into open list and parent is put into closed list.
- Assumption: Monotone restriction is satisfied. That is the estimated cost of reaching the goal node for a particular node is no more than the cost of reaching a child and the estimated cost of reaching the goal from the child



$$h(n_1) \leq C(n_1, n_2) + h(n_2)$$



BFS, DFS – Uninformed / Brute Force Search methods

Algorithm A

- A function f is maintained with each node

  $f(n) = g(n) + h(n)$, n is the node in the open list

- Node chosen for expansion is the one with least f value
- For BFS: $h = 0$, g = number of edges in the path to S
- For DFS: $h = 0$, $g = \dfrac{1}{\text{No of edges in the path to S}}$

Algorithm A*

- One of the most important advances in AI
- *g(n)* = least cost path to n from S found so far
- *h(n)* <= *h*(n)* where *h*(n)* is the actual cost of optimal path to G(node to be found) from *n*

"Optimism leads to optimality"