

Procedure GRAPHSEARCH

- 1 Create a *search graph*, G , consisting solely of the start node, s . Put s on a list called *OPEN*.
- 2 Create a list called *CLOSED* that is initially empty.
- 3 LOOP: if *OPEN* is empty, exit with failure.
- 4 Select the first node on *OPEN*, remove it from *OPEN*, and put it on *CLOSED*. Call this node n .
- 5 If n is a goal node, exit successfully with the solution obtained by tracing a path along the pointers from n to s in G . (Pointers are established in step 7.)
- 6 Expand node n , generating the set, M , of its successors that are not ancestors of n . Install these members of M as successors of n in G .
- 7 Establish a pointer to n from those members of M that were not already in G (i.e., not already on either *OPEN* or *CLOSED*). Add these members of M to *OPEN*. For each member of M that was already on *OPEN* or *CLOSED*, decide whether or not to redirect its pointer to n . (See text.) For each member of M already on *CLOSED*, decide for each of its descendants in G whether or not to redirect its pointer. (See text.)
- 8 Reorder the list *OPEN*, either according to some arbitrary scheme or according to heuristic merit.
- 9 GO LOOP