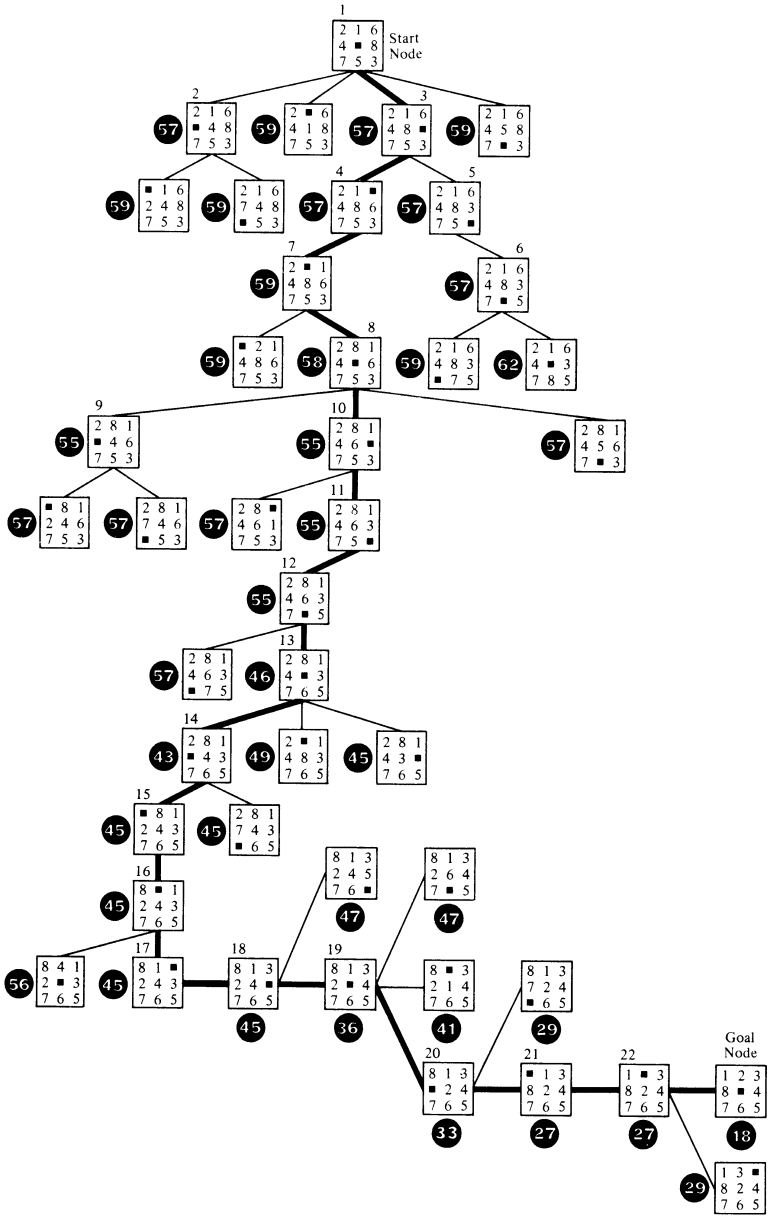


*Fig. 2.9 A search tree for the 8-puzzle.*





Again, the $f$ values of each node are circled in the figure, and the uncircled numbers show the order in which nodes are expanded. (In the search depicted in Figure 2.9, ties among minimal $f$ values are resolved by selecting the deepest node in the search tree.)

The solution path found happens to be of minimal length (18 steps); although, since the $h$ function is not a lower bound for $h^*$, we were not guaranteed of finding an optimal path. Note that this $h$ function results in a focused search, directed toward the goal; only a very limited spread occurred, near the start.

Another factor that determines the heuristic power of search algorithms is the amount of effort involved in calculating the heuristic function. The best function would be one identically equal to $h^*$, resulting in an absolute minimum number of node expansions. (Such an $h$ could, for example, be determined as a result of a separate complete search at every node; but this obviously would not reduce the total computational effort.) Sometimes an $h$ function that is not a lower bound on $h^*$ is easier to compute than one that is a lower bound. In these cases, the heuristic power might be doubly improved—because the total number of nodes expanded can be reduced (at the expense of admissibility) and because the computational effort is reduced.

In certain cases the heuristic power of a given heuristic function can be increased simply by multiplying it by some positive constant greater than one. If this constant is very large, the situation is as if $g(n) \equiv 0$. In many problems we merely desire to find *some* path to a goal node and are unconcerned about the cost of the resulting path. (We are, of course, concerned about the amount of search effort required to find a path.) In such situations, we might think that $g$ could be ignored completely since, at any stage during the search, we don't care about the costs of the paths developed thus far. We care only about the remaining seach effort required to find a goal node. This search effort, while possibly dependent on the $h$ values of the nodes on *OPEN*, would seem to be independent of the $g$ values of these nodes. Therefore, for such problems, we might be led to use $f \equiv h$ as the evaluation function.

To ensure that *some* path to a goal will eventually be found, $g$ should be included in $f$ even when it is not essential to find a path of minimal cost. Such insurance is necessary whenever $h$ is not a perfect estimator; if the node with minimum $h$ were always expanded, the search process might expand deceptive nodes forever without ever reaching a goal node.