1 Answers

1. no
2. yes
3. yes, no, yes, yes
4. BFS won’t terminate
5. Assuming you mark nodes, $O(n)$ in all cases. If you don’t mark nodes, then assuming there’s a solution, BFS is worst-case $O(n^n)$. If there isn’t a solution, or you use DFS without marking, it could take forever.
6. -
7. (a) queue (b) stack (c) priority queue (d) priority queue
8. Uniform cost; the specialization is that Dijkstra’s is graph search (as opposed to tree search).
9. -
10. -
11. -
12. No, no.
14. When the heuristic is admissible.
15. -
16. When the heuristic is consistent.
17. It makes it somewhat simpler to write A*.
18. Any of the heuristics discussed for constraint satisfaction problems, for instance the “minimum remaining values” (MRV) heuristic. Notice that heuristics like MRV are not heuristic functions $h(x)$; the word “heuristic” here means something more like “rule of thumb.” That being said, it may be possible to get the effect of MRV with a suitably defined informed search problem. An example of an honest-to-goodness heuristic function $h(x)$ which might be useful when treating Sudoku as a search problem is the following: let $h(s)$ be the number of squares whose values can’t be determined by running an arc consistency checker on the state $s$.
19. Use a binary heap for the priority queue.
20. A chain of $n$ nodes.