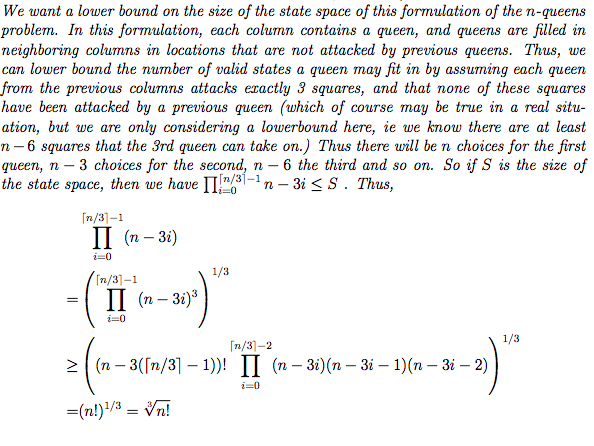
*3.6*

*../Desktop/螢幕快照%202018-11-21%2018.14.11.png*

*3.15*

*a.* FALSE. Depth-first search may possibly, sometimes, BY GOOD LUCK, expand fewer nodes than A\* search with an admissible heuristic. E.g., it is logically possible that sometimes, by good luck, depth-first search may march directly to the goal with no back-tracking.

*b.*TRUE. h(n)=0 NEVER over-estimates the remaining optimal distance to a goal node.

*c.FALSE.* The continuous spaces can be discretized. A\* (or variants) are widely used, e.g., for navigation.

*d.* TRUE, because if there exists a goal it occurs at finite depth *d* and will be found in *O(bd)* steps. “Complete” means “will find a goal when one exists” --- and does NOT imply “optimal,” which means “will find a lowest-cost goal when one exists.” Thus, the step costs are irrelevant to “complete.”

e.FALSE. The Manhattan distance may over-estimate the optimal remaining number of moves to the goal because a rook may cover several squares in a single move. If the path cost instead were the number of squares covered, then Manhattan distance would be admissible.

3.22

a. When all step costs are equal (and let’s assume equal to 1), g(n) is just a multiple of depth n. *f(n)* = *g(n)=depth(n)* Thus, breadth-first search and uniform-cost search would behave the same in this case

b. setting *f(n)* = *g(n)=-depth(n)* thus forcing deep nodes on the current branch to be searched before shallow nodes on other branches

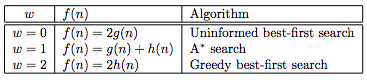
c. A\* search: f(n) = g(n) + h(n)

Uniform-cost search: f(n) = g(n)

for h(n) = 0, uniform cost search will produce the same result as A\* search

3.28

The algorithm is guaranteed to be optimal for 0 ≤ w ≤ 1, since scaling g(n) by a constant has no effect on the relative ordering of the chosen paths, but, if w > 1 then it is possible the wh(n) will overestimate the distance to the goal, making the heuristic inadmissible. If w ≤ 1, then it will reduce the estimate, but it is still guaranteed to underestimate the distance to the goal state.



3.29

a. The branching factor is 4 (number of neighbors of each location).

b. The states at depth k form a square rotated at 45 degrees to the grid. Obviously there

are a linear number of states along the boundary of the square, so the answer is 4k.

c. Without repeated state checking, BFS expends exponentially many nodes: counting

precisely, we get ((4x+y+1 − 1)/3) − 1.

d. There are quadraticallymany stateswithin the square for depth x + y, so the answer is

2(x + y)(x + y + 1) − 1.

e. True; this is theManhattan distancemetric.

f. False; all nodes in the rectangle defined by (0, 0) and (x, y) are candidates for the

optimal path, and there are quadratically many of them, all of which may be expended

in the worst case.

g. True; removing linksmay induce detours, which requiremore steps, so h is an underestimate.

h. False; nonlocal links can reduce the actual path length below the Manhattan distance.

3.32

A heuristic is consistent iff, for every node n and every successor n′ of n generated by

any action a,

h(n) ≤ c(n, a, n′) + h(n′)

One simple proof is by induction on the number k of nodes on the shortest path to any goal from n. For k = 1, let n′ be the goal node; then h(n) ≤ c(n, a, n′). For the inductive

case, assume n′ is on the shortest path k steps from the goal and that h(n′) is admissible by hypothesis; then

h(n) ≤ c(n, a, n′) + h(n′) ≤ c(n, a, n′) + h∗(n′) = h∗(n)

so h(n) at k + 1 steps from the goal is also admissible.