**CSC 411 Project 1**

Task 1:

**The DFS trace**

1. Open = [A]; closed = [ ]
2. Open = [B, C, D]; closed = [A]
3. Open = [E, F, C, D]; closed = [B, A]
4. Open = [K, L, F, C, D]; closed = [E, B, A]
5. Open = [S, L, F, C, D]; closed = [K, E, B, A]
6. Open = [L, F, C, D]; closed = [S, K, E, B, A]
7. Open = [T, F, C, D]; closed = [L, S, K, E, B, A]
8. Open = [F, C, D]; closed = [T, L, S, K, E, B, A]
9. Open = [M, C, D]; closed = [F, T, L, S, K, E, B, A]
10. Open = [C, D]; closed = [M, F, T, L, S, K, E, B, A]
11. Open = [G, H, D]; closed = [C, M, F, T, L, S, K, E, B, A]
12. Open = [H, D]; closed [C, M, F, T, L, S, K, E, B, A]
13. Open = [O, P, D]; closed [C, M, F, T, L, S, K, E, B, A, H]
14. Open = [P, D]; closed [C, M, F, T, L, S, K, E, B, A, H, O]
15. Open = [U, D]; closed [C, M, F, T, L, S, K, E, B, A, H, O, P]
16. Open = [D]; U is found goal state reached, while loop does not continue

Task 2:

**BFS Pseudocode**

Begin

Open := [Start] **// initialize**

Closed := [];

While open != [] do **// states remain**

Begin

Remove leftmost state from open, Call it X;

If X is a goal then return SUCCESS **// goal found**

Else begin

Generate children of X;

Put X on closed;

Discard children of X if already on open/closed **// check**

Put remaining children on right end of open **// que**

End

End

Return FAIL **// no states remain**

end

**DFS Pseudocode**

Begin

Open := [Start] **// initialize**

Closed := [];

While open != [] do **//states remain**

Begin

Remove leftmost state from open, call it X;

If X is a goal then return SUCCESS **// goal found**

Else begin

Generate children of X;

Put X on closed;

Discard children of X if already on open or closed; **//check**

Put remaining children on left end of open **//stack**

End

End;

Return FAIL **// no states left**

End

Task 3:

After running the program with different start positions it seems that DFS works better. The criteria I use to deduce this is that it finds an optimal path in less time and visits less nodes, thus using less memory.