1. Download tromp.c (or copy it from simple/go in the class github repo) and execute it. You should get total 1446 as part of the output.

Now modify tromp.c: move the two consecutive lines that begin s = passed ?... and if (s > alpha... to after the for loop (but before the return). Re-compile and re-execute. What value do you get for total?

2. A move by a player is silly if some alternate move either leads to a better final score or leads to a shorter game (and same or better final score). The link below shows a silly 2x2 game with 46 moves: Explain why move 3 in this 2x2 game is silly.
https://senseis.xmp.net/?LongestPossibleGame

3. For each of the following sliding tile positions, (i) give the total number of inversions (ii) use the parity rule to explain whether it is solvable,

```
1 3 2 . 5 3 2 5 3 . 7 2 6 3 . 2 6 3
2 . 1 3 1 4 . 4 1 2 . 4 5 1 7 4 1 5
```

4. Prove that the sliding tile puzzles below are either both solvable, or both unsolvable. Hint: show that the parity rule gives the same answer for each puzzle.

```
15 13 11 12 15 13 11 12
8 4 9 3 8 4 9 3
1 7 6 2 1 7 . 2
5 10 . 14 5 10 6 14
```

5. Let \( P \) be any sliding tile puzzle with at least 2 rows and at least 2 columns. Let \( P' \) be the sliding tile puzzle obtained from \( P \) by (warning: this is not a legal move!) swapping the rightmost two tiles in the bottom row. Prove that \( P \) is solvable if and only if \( P' \) is not solvable. Hint: use the parity formula.

6. Using the previous question, for the sliding tile puzzle \( r \geq 2 \) rows and \( c \geq 2 \) columns, explain why exactly one half of all positions are solvable.

7. The webnotes shows the top 4 levels of the search tree from the position below left. Draw the top 4 levels of the search tree from the position below right.

```
2 3 5 4 * 1
4 1 * 5 2 3
```

8. Recall that the 2x3 sliding tile state space graph \( G_{2,3} \) has a total of \((2 \times 3)! = 720\) nodes and that the average node degree (number of neighbours) is \((2 + 3 + 2 + 2 + 3 + 2)/6 = 7/3\). The number of edges in a graph with \( n \) nodes and average degree \( d \) is \( n \times d/2 \), so \( G_{2,3} \) has 840 edges.

(i) Give the number of nodes and edges in \( G_{3,3} \) and \( G_{4,4} \).

(ii) Assume that breadth-first-search on a graph \( G \) with \( n \) nodes and \( m \) edges runs in time proportional to \((n + m)\). Assume that BFS of \( G_{2,3} \) takes 1 second, how long will a BFS of \( G_{3,3} \) take? Of \( G_{4,4} \)? Show your work.