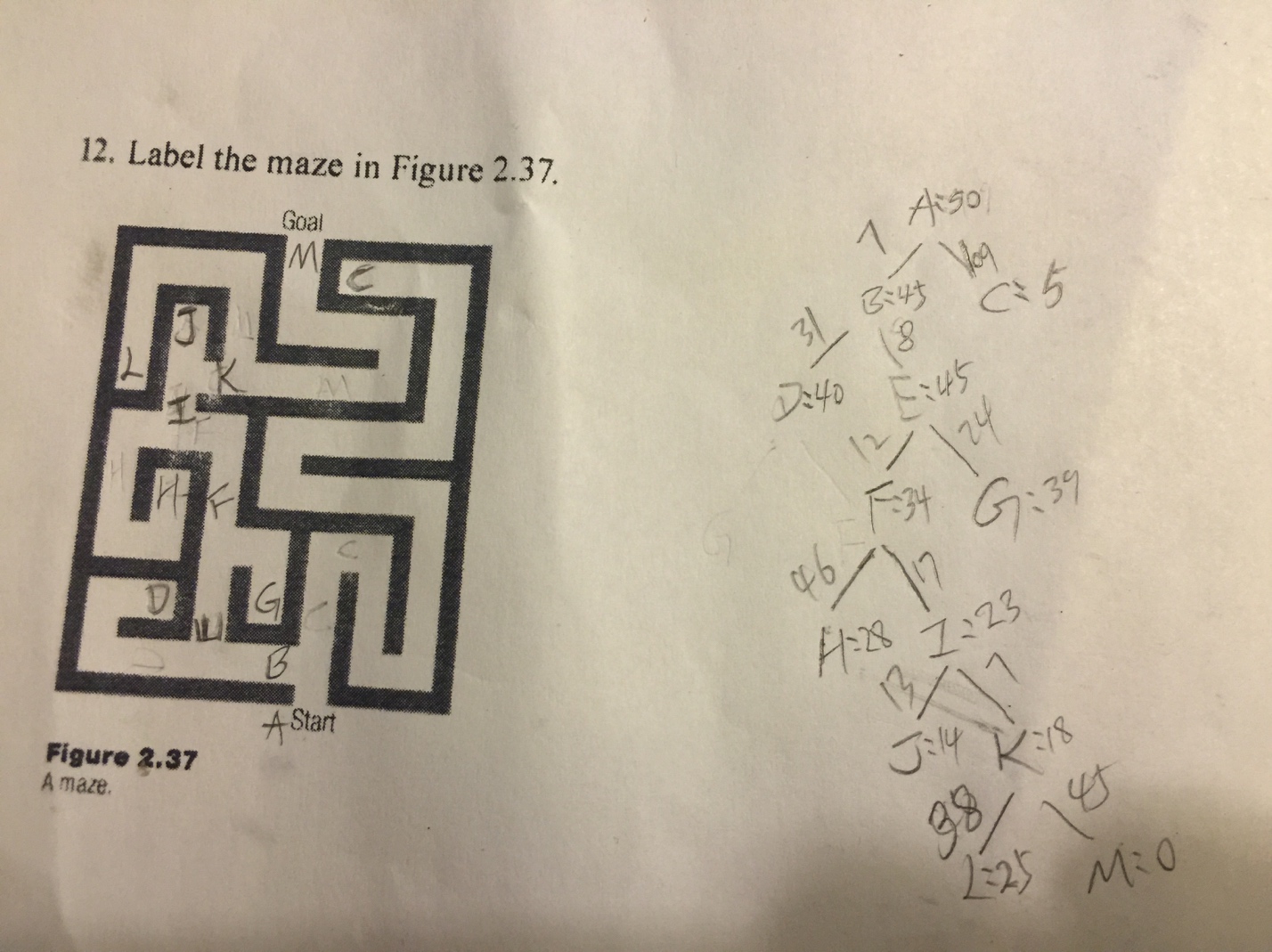
Program1 Report



1. Hill Climbing Search:

This is the first informed search algorithm. The steepest trail will always be chosen. However, it is not possible to return previous altitude when stuck. In our case, the search is stuck at node C.

1. Best First Search:

It is an informed search using heuristic. It may get stuck in an infinite loop and not optimal. In this assignment, it got stuck. B->D , D->B infinite loop

Worst time complexity:





1. Beam Search:

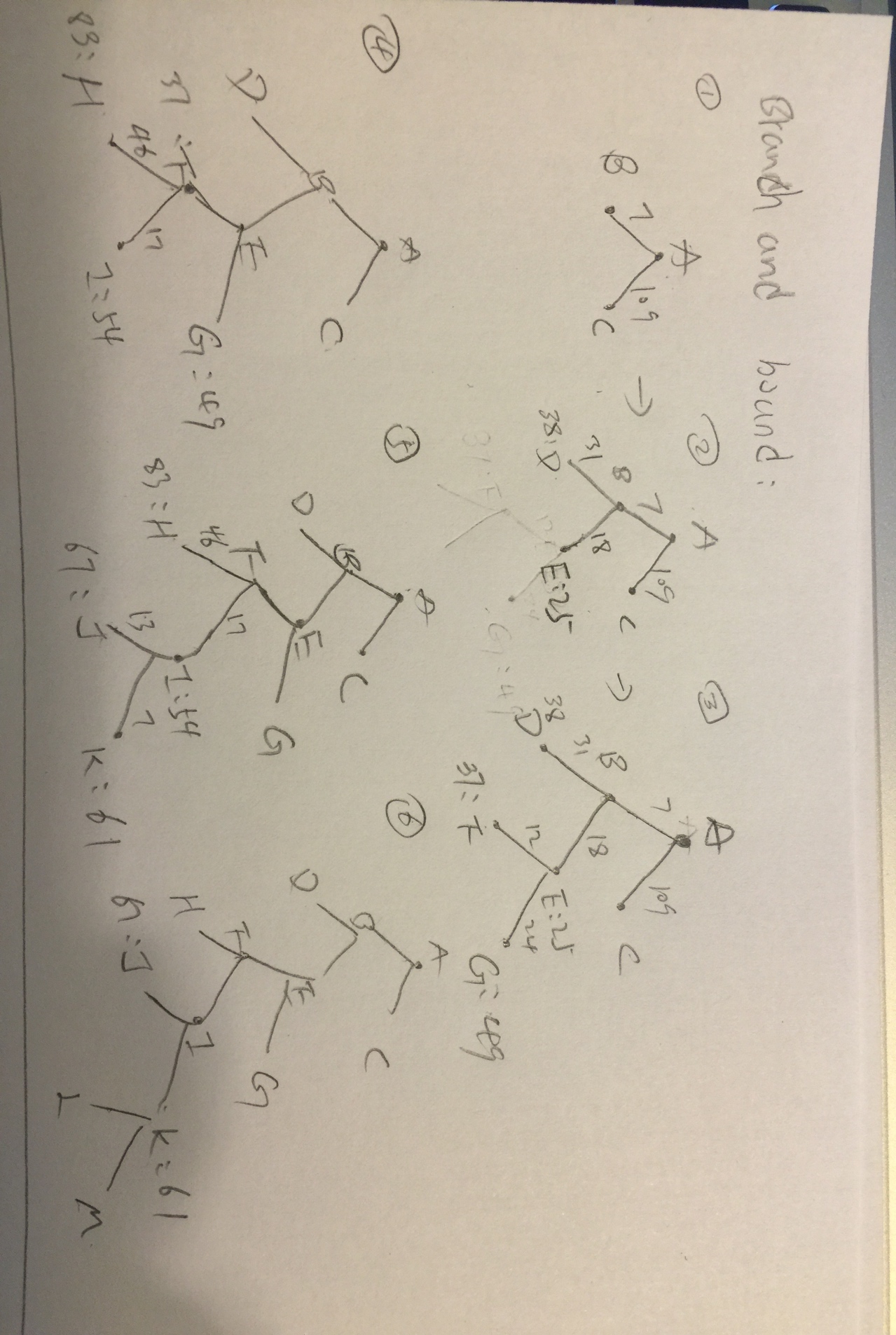
This is another informed search with Heuristics. Since a goal state could potentially be pruned, beam search is not complete (the guarantee that an algorithm will terminate with a solution, if one exists). Beam search is not optimal (that is, there is no guarantee that it will find the best solution). It returns the first solution found. In this case, W =2. Investigation spread level by level and only the best w nodes are expanded. Beam-search worst time complexity with width k O(kd)

After run the beam search on this maze, I got the path ['A', 'B', 'E', 'F', 'I', 'K', 'M']

Branch and Bound Search(plain vanilla)

This search doesn’t use heuristic estimate, it uses only the real cost in the path

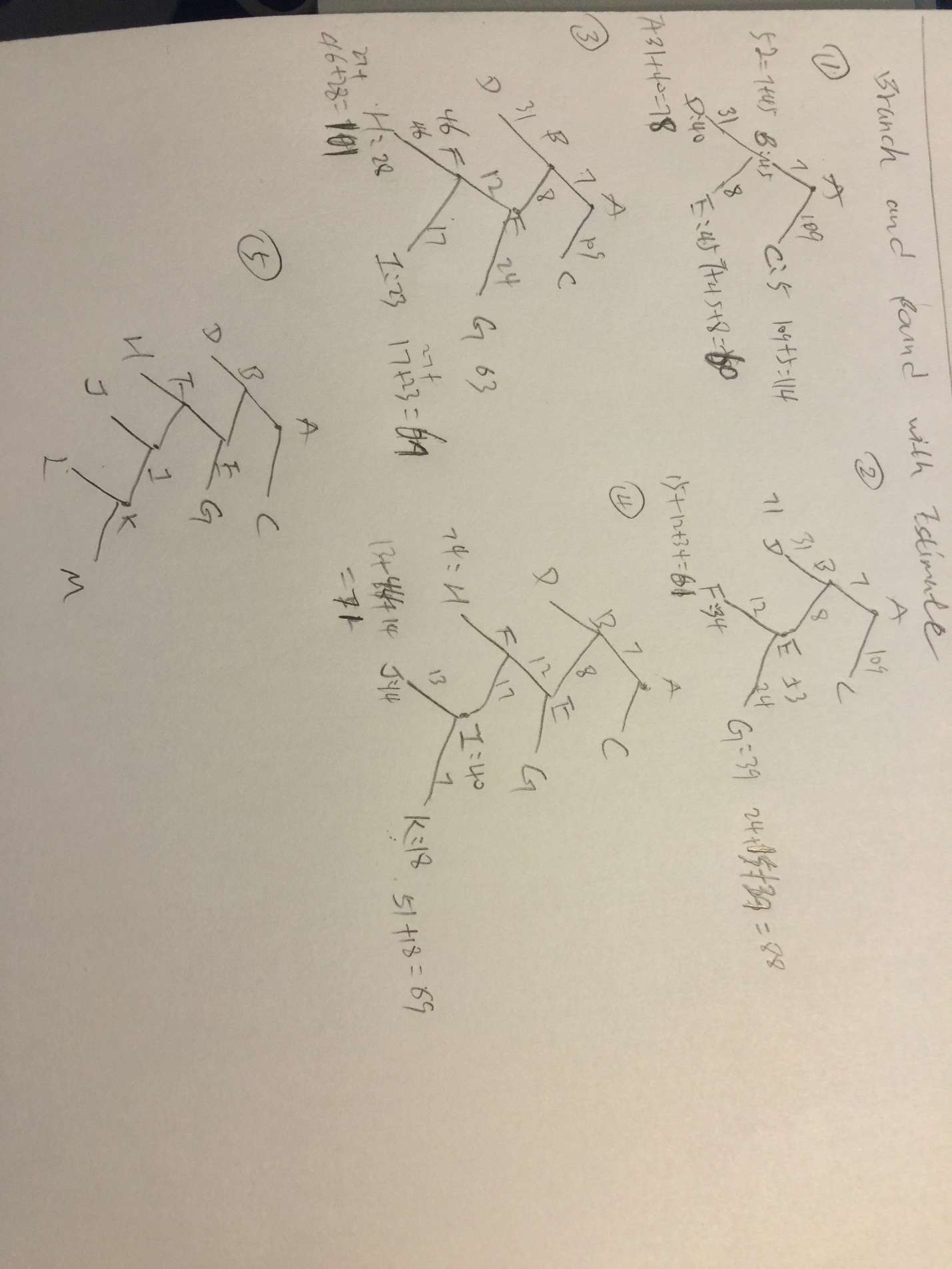
The result is: ['A', 'B', 'E', 'F', 'I', 'K', 'M'] after running. Below attached is the steps to perform this search:



1. Branch and Bound Searchwith underestimate:

This search uses Heuristic estimate and real cost

The result is: ['A', 'B', 'E', 'F', 'I', 'K', 'M'] after running. Below attached shows the steps to perform this search:



1. Branch and Bound with Dynamic Programming:

This search uses **Principle of Optimality** – optimal paths are constructed from optimal subpaths. i.e. an optimal subpath from S to G that passes through some intermediate node I is composed of an optimal S -> I path, followed by an optimal I -> G path. “ If two or more paths reach a common node, only the path that reaches this common node with the minimum cost should be stored. (Delete the others!) ”

The result is: ['A', 'B', 'E', 'F', 'I', 'K', 'M'] after running

1. A\* search:

This search program employs branch and bound with both estimates of remaining distance and dynamic programming. It is complete and optimal.

The result is: ['A', 'B', 'E', 'F', 'I', 'K', 'M'] after running