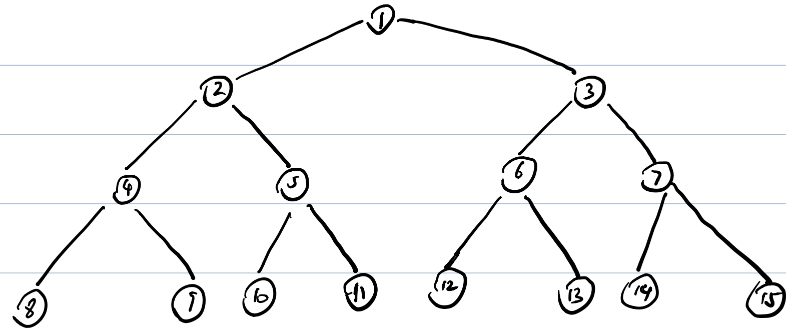
1. (a) Solution:



1. (b) Solution:

Breadth-first search: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11

Depth-limited search: 1, 2, 4, 8, 9, 5, 10, 11

Iterative-deepening search: 1; 1, 2, 3; 1, 2, 4, 5, 3, 6, 7; 1, 2, 4, 8, 9, 5, 10, 11

1. (c) Solution:

Bidirectional search could help reduce time complexity and concentrate the search. In this model, the branching factor in the forward direction is 2, while the branching factor in the reverse direction is 1.

1. (d) Solution:

Yes. We could start from the goal state and apply the reverse successor action until we reach 1.

1. (e) Solution:

Yes. We notice that 2k and 2k+1 is a binary expansion. If we start from k and go Left, we get 2k, otherwise we step right and get 2k+1. Hence if goal is 11 which in binary is 1011, we could use Left, Right Right.

2. (a) Solution:

If we are ignoring the one-per-square constraint, the n vehicles and n^2 locations correspond to $(n^2)^n = n^{2n}$ states.

If we take the one-per-square constraint into consideration, the n vehicles and n^2 locations correspond to $\binom{n^2}{n}$ states.

2. (b) Solution:

As we know, on each time step, each vehicle could have 5 different actions. Hence the branching factor of n is 5^n.

2. (c) Solution:

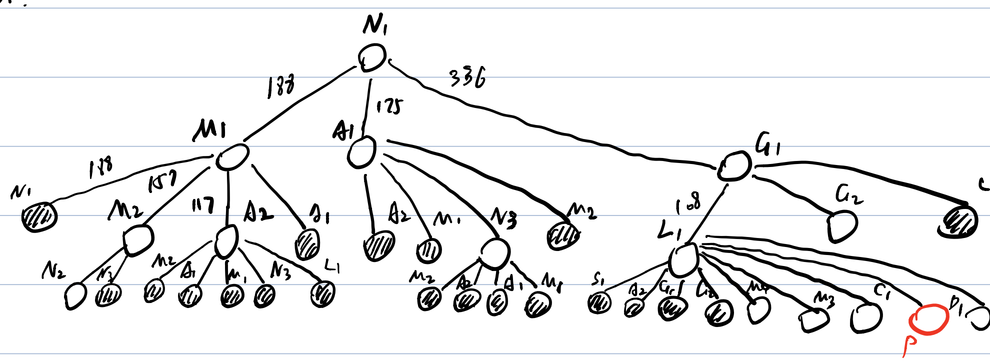
We construct our heuristic function by implementing Manhattan Distance.

$h_i = |(n-i+1) - x_i| + |n - y_i|$

2. (d) Solution:

Only (iii) works. According to the definition of admissible heuristic:
an admissible heuristic is the one that never overestimate the cost to reach the goal. First, set T to be the total distance moved by all vehicles in a given solution. We have $T \geq \sum_i h_i \geq n \cdot \min\{h_1, \ldots h_n\}$.
Second, every step we could at most get n works done. Hence, completing all the work requires at least $n \cdot \min\{h_1, \ldots h_n\} / n = \min\{h_1, \ldots h_n\}$ steps.

3. a) Solution:

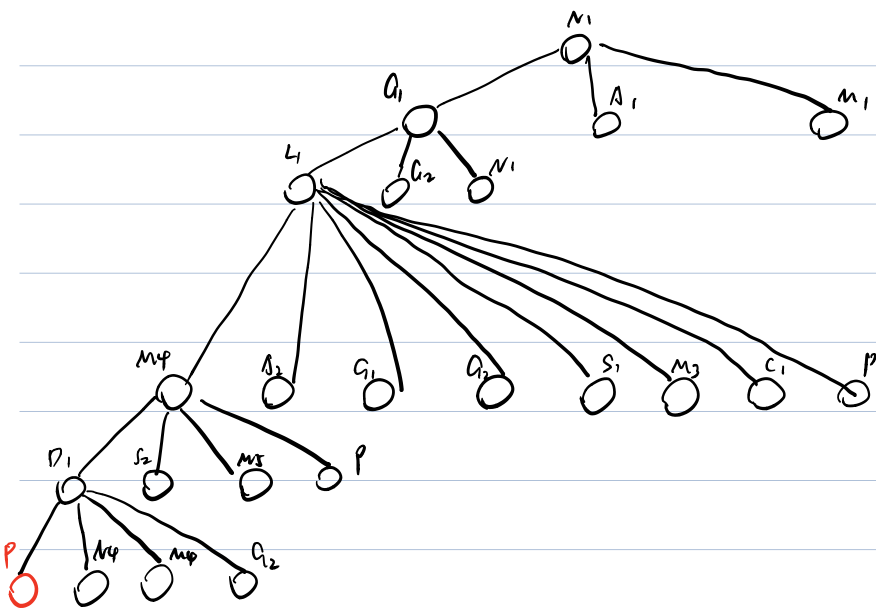


Route: Nice → Grenoble → Lyon → Paris

The solution is not optimal

Total route cost: 336 + 108 + 451 = 895

(b) Solution:

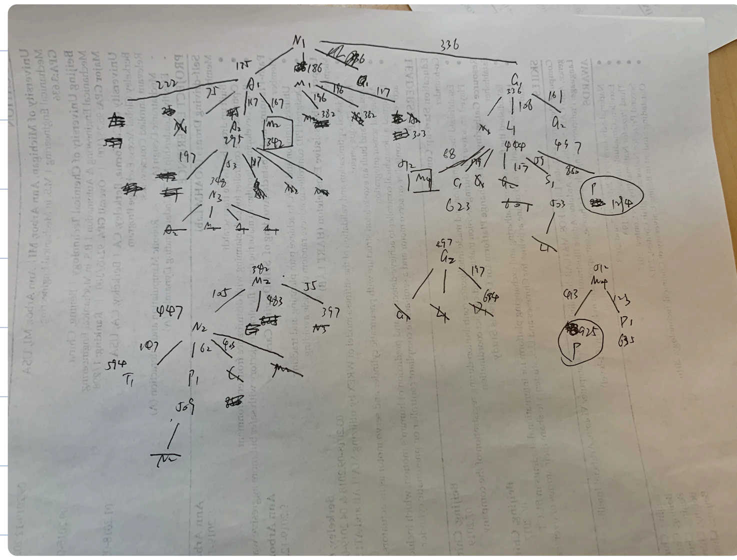


Route: Nice → Grenoble → Lyon → Mâcon → Dijon → Paris

Total route cost: 336 + 108 + 88 + 123 + 311 = 966.

The solution is not optimal.

(c) Solution :

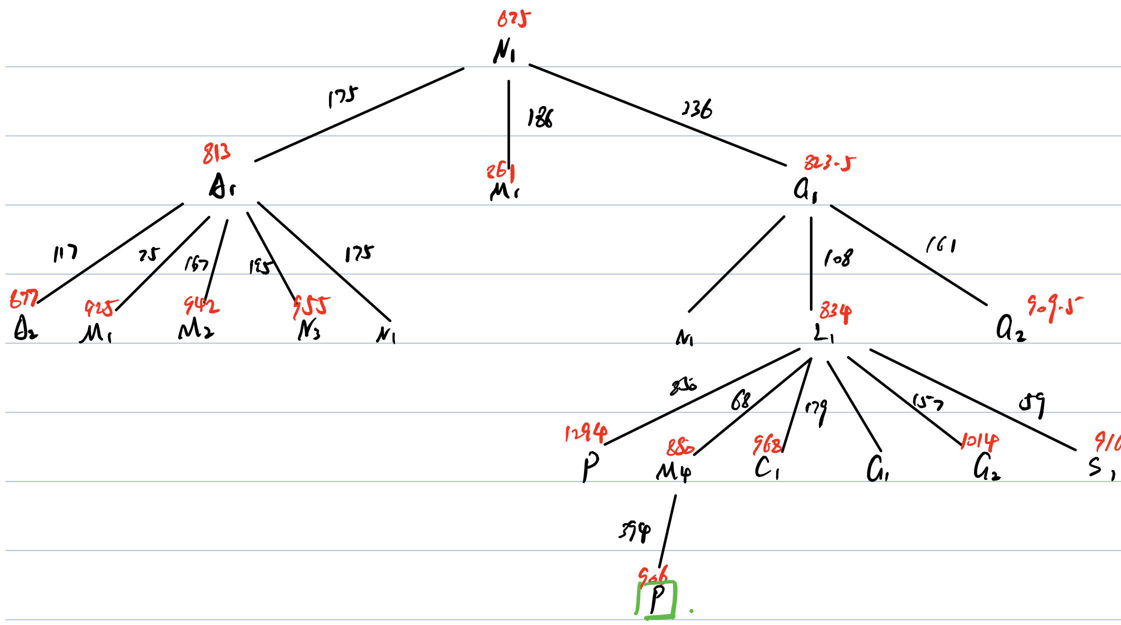


Route : Nice -> Grenoble -> Lyon -> Mâcon -> Paris

Total cost : 336 + 108 + 68 + 413 = 925

The solution is optimal.

(d) Solution:



Route: Nice → Grenoble → Lyon → Mâcon → Paris.

Total cost: 906

The solution is optimal.

M_1 = Marseille	N_1 = Nice	M_4 = Mâcon	S_2 = Strasburg
M_2 = Montpellier	N_2 = Narbonne	D_1 = Dijon	M_5 = Mulhouse
$A1$ = Aix	N_3 = Nîmes	P_1 = Perpignan	
A_2 = Avignon	L_1 = Lyon	T_1 = Toulouse	
G_1 = Grenoble	S_1 = St-Etienne	C_1 = Clermont-Ferrand.	
G_2 = Genève	M_3 = Montluçon	N_4 = Nancy	