1. Given the following 0-1 knapsack problem with $W = 13$kg:

i	v_i	w_i
1	\$20	2kg
2	\$30	5kg
3	\$35	7kg
4	\$12	3kg
5	\$3	1kg

Draw the pruned state space tree and show the actions step by step using:

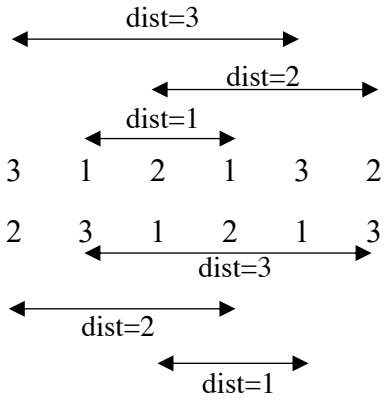
(a) Depth-first search (15 marks).
(b) Breadth-first search (15 marks).
(c) Best-first search (15 marks).

2. Consider the following problem. Given a positive number N, find all combinations of $2N$ elements with the following conditions:
   - Every element from 1 to N appears exactly twice.
   - The distance between the same element is exactly the value of the element.

   For example:

   Input: $N = 3$

   Output:



   More examples:

   Input: $N = 4$

   Output

   4 1 3 1 2 4 3 2

   2 3 4 2 1 3 1 4

Input: $N = 7$

Output:

1 7 1 2 5 6 2 3 4 7 5 3 6 4

5 1 7 1 6 2 5 4 2 3 7 6 4 3

4 1 7 1 6 4 2 5 3 2 7 6 3 5

…

…

Total 52 possible combinations.

(a) Design the state space tree and describe your idea to solve this problem by backtracking. (25 marks)
(b) Implement this algorithm in Python as a function:

    def twice_distance(n):

    The sum of running time is recorded. The fastest three implementations in the class will get a bonus 10 marks for this assignment. (30 marks)