Student who wrote the solution to this question: Zimo Li, Yu-Hsuan Chuang, Jiahong Zhai
Students who read this solution to verify its clarity and correctness: Zimo Li, Yu-Hsuan Chuang, Jiahong Zhai

Algorithm for 3.a 3.b together:
First need to initialize $C(n \times n)$ matrix for storing cost of moving cars for part a, and $S(n \times n)$ matrix for storing the index of split car when joining a segment for part b. We define tsqrt(i, j) as the square root of the total weight of cars i to j add together.

Pseudocode:

```
main(1,n): // O(n^3)
    for l in 1~n:
        for i in 1~(n-l+1):         // O(n^2) inner loop iterations
            j = i + l - 1
            DP(i,j)         // O(n) each time
    return C(1,n)

DP(i,j): // O(n)
    if i == j:
        C(i,i) = tsqrt(i,i)
        S(i,i) = i
    else:
        s = i
        cost = +inf
        for k in i~j-1: // O(n) loop iterations
            if min{tsqrt(i,k), tsqrt(k+1,j)} + C(i,k) + C(k+1,j) < cost:
                cost = min{tsqrt(i,k), tsqrt(k+1,j)} + C(i,k) + C(k+1,j)
                s = k
        C(i,j) = cost
        S(i,j) = s
```

(a) To find best cost of forming car segment i j, need to split the segment to 2 part. Need to find all possibility of splitting and their cost respectively, the minimum cost of joining these two parts plus their own cost is what we want. Because every join operation actually join two segments. As for the actual cost of i j, C(i,j):

$$C(i,j) = \begin{cases} 0, & \text{if } i = j \\ \min\limits_{i<=k<j} \left[\min\left(tsqrt(i,k),\ tsqrt(k+1,j) \right) + C(i,k) + C(k+1,j) \right], & \text{if } i! = j \end{cases}$$

The pseudo code is given above.
Run time is $O(N^3)$. Because main function iterate $O(n^2)$ times, each time calls DP once, and DP iterate $O(n)$ times. So in combine, complexity is $O(N^3)$.

(b) In the above pseudo code, every time an entry in C matrix is confirmed, then the corresponding entry indicating the index of splitting train car in S matrix is also stored. To print out an optimal order of train car join operation from i to j, use the pseudo-code below:

```
PrintOrder(i,j):
    if(i = j):
       do nothing     //no need to connect one car to itself
    else
       print("segment from car %d to car %d split by car %d", i, j, S(i,j))
       PrintOrder(i, S(i,j))
       PrintOrder(S(i,j)+1, j)
```