Se sortează descrescător dupa capacitate / greutate atât traseele cât şi camioanele. Traseul de capacitate maximă va fi parcurs de camionul cu greutatea cea mai mare (dintre camioane) dar care este mai mică sau egală cu capacitatea traseului. După sortare, acesta este primul (de la stânga la dreapta) ce are greutatea admisă. Apoi se tratează al doilea traseu similar, cel de capacitate mai mică şi care va fi parcurs de următorul camion cu greutate cât mai mare (dintre camioane), dar suficient de mică (relativ la traseu). Poziţia acestui camion în vectorul sortat este strict la dreapta faţă de camionul anterior, astfel indicele care parcurge vectorul camioanelor se deplasează doar spre dreapta şi complexitatea se *amortizează* (deşi sunt doua structuri repetitive imbricate, dimensiunea lor nu se înmulţeşte ci se adună). Evident dacă s-a ajuns la capătul vectorului de camioane înseamnă că nu se mai pot asigna camioane la trasee. La final traseele trebuie afişate în ordinea iniţială. Algoritmul are complexitate O(N+M) după sortare, şi O(NlogN + MlogM) în total, dacă se foloseşte o sortare eficientă de exemplu cea din STL.

Pentru punctaj maxim, trebuie avut în vedere faptul că numerele nu încap nici pe 32 biti (int) şi nici pe 64 (long long), astfel încât trebuie implementate operaţii pe numere mari (şiruri de caractere ce reprezintă cifre), din fericire este suficientă implementarea doar operatorului de comparare.

Corectitudinea algoritmului descris se poate demonstra matematic.

Diferite punctaje parţiale se obţin atât pentru implementarea pe int, long long, cât şi pentru algoritmi de complexitate mai mare ca O(NxM), sau chiar backtracking O(N!).