b. What is the run time complexity of your implementation.

Output of mincost function below ,

From 0 to 1 cost is 3 Complexity=1
From 0 to 2 cost is 5 Complexity=3
From 0 to 3 cost is 7 Complexity=9
From 0 to 4 cost is 8 Complexity=27
From 0 to 5 cost is 6 Complexity=81
From 0 to 6 cost is 7 Complexity=243

According to this complexity values shows $T(n)=3^n$ functionality here n is the destination station from 0^{th} station and T(n) is runtime complexity so this order of exponential so complexity of mincost function is $O(n^n)$ so which is exponential .

c. Argue that dynamic programming can be used to improve the runtime.

Dynamic programming is useful is mincost recursive algorithm finds itself reaching the same situations (input parameters) many times.
There is a general transformation from recursive algorithms to dynamic programming known as memoization,
In dynamic programming there is a table storing all results ever calculated by our recursive procedure.
When the recursive procedure is called on a set of inputs which were already used, the results are just fetched from the table.
This reduces recursive algorithm time compexity
Dynamic programming can be even smarter, applying more specific optimizations.

e.Calculate the runtime of your implementation in part 4 above. Assume, hashing is O(1).

Output of the mincostDynamicpro function below ,
From 0 to 1 cost is 3Complexity=1
From 0 to 2 cost is 5Complexity=3
From 0 to 3 cost is 7Complexity=7
From 0 to 4 cost is 8Complexity=13
From 0 to 5 cost is 6Complexity=21
From 0 to 6 cost is 7Complexity=31

According to this complexity values shows $T(n)=n^2+n+1$ functionality here n is the destination station from 0^{th} station and T(n) is runtime complexity so this is order of quadratic therefor complexity of mincostDynamicpro function is $O(n^2)$ which is quadratic.