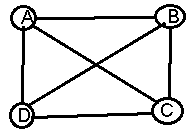
**Problem 2**

From HC instance G = (V, E), adds missing edges to G to produce a complete graph H = (VH, EH), and then define c: H N by c(e) = 0 if e E, but c(e) = 1 if e E. Finally, let k = 0. Our TSP instance is (H, c, k).



1. So now solution for HC —> will be the solution for TSP instance

2. Solution for TSP —> will be the solution for HC instance

**Problem 4**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| N[0] |  |  |  |  |  |  |  |  |  |  |  |
| N[1] |  |  |  |  |  |  |  |  |  |  |  |
| N[2] |  |  |  |  |  |  |  |  |  |  |  |
| … |  |  |  |  |  |  |  |  |  |  |  |
| N[n] |  |  |  |  |  |  |  |  |  |  |  |

O (n) algorithm

* Using the memorization table for the SubsetSum problem given by N array and the number sum k = 10
* So the table will be filled from row N[0] to N[N.length - 1]
* Continue that way and look at the cell to see if there is any cell that contains the number whose sum is 10. Then we found the solution for it
* The running time to fill in to the data table is O(kn) which is O(10n) —> O(n)