Programming Report #1

1015087학번

컴퓨터 공학과

이기연

* **문제 기술:**

이번 과제는 TSP(Traveling Salesman Problem) 문제에 대해 수업시간에 배운 다양한 알고리즘 방법을 적용하여 프로그래밍하는 과제이다. 이 문제의 경우에는 optimal solution을 찾기 힘들어 다양한 알고리즘은 적용한 것이다. Divide-and-Conquer TSP, Dynamic programming TSP, Greedy TSP의 알고리즘을 프로그래밍하였다.

* **입출력의 예:**

INPUT:

**Number of cities =** 가 뜨면 여기에 n 즉, 총 몇개의 노드를 할 지 입력한다.

OUTPUT:

**Divide-and-Conquer TSP: ~**

**Dynamic programming TSP: ~**

**Greedy TSP: ~**

각 해당하는 값이 나온다.

* **문제풀이 방법(알고리즘) 기술:**

Divide-and-Conquer TSP: 리컬시브하게 프로그램을 돌려보았다.

Dynamic programming TSP: subset관련해서 배열을 새로 만들어 값을 저장했다.

Greedy TSP: root와 저장되어있는 것을 조사하여 넣는 배열을 만들었다.

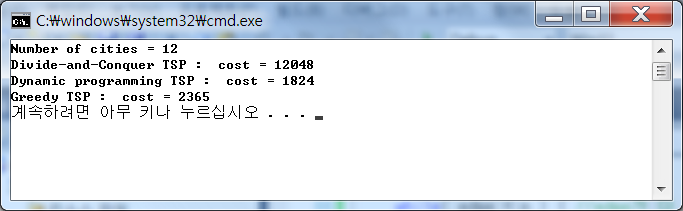
* **소스 프로그램:**

변수명의 의미, 각 단계별 기술, 부프로그램의 기능에 대한 설명이 소스프로그램 내에 commend로 작성되어 있다.

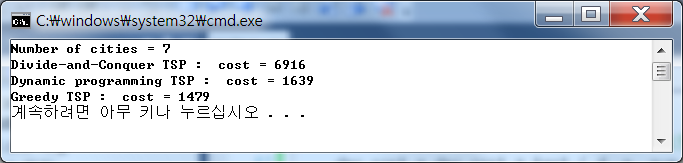
* **수행 결과:**

Input: 12

Output:

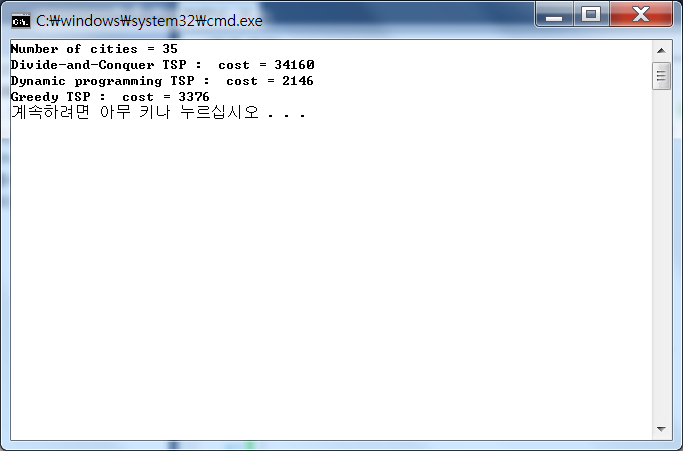


Input: 7

Output: 

Input: 35

Output:



* **결과 분석 및 토의:**

이번 프로그램을 하면서Divide-and-Conquer TSP은 잘 못 프로그래밍한다면 무한 루프에 빠질 확률이 높았고, 어느 부분을 잘라줘서 리컬시브하게 돌려야하는지 그게 참 고민을 많이 하게 하였다.

Dynamic programming TSP관련해서는 수도코드를 보면서 너무나도 subset에 고민을 많이 하여 이걸 어떻게 풀까만 고민하였는데, 다이나믹 프로그램의 원리 즉, 리컬시브하게 돌리는 것이 아니라 구한 값을 배열에 저장하여 그걸 다시 쓰는 원리를 생각하여 subset을 나타내주기만 하는 배열을 만들어 그 곳에 값을 저장하여 프로그램을 작성해보았다.

Greedy TSP는 의외로 간단하게 min값과 과제에서 주어진 2개의 조건문만 만족시키면 되므로 프로그램 중에 가장 간단히 작성하였다.

아마도 Divide-and-Conquer(이하 DAC) TSP 코드에 문제가 있는 것 같아 보였다. Dynamic과 Greedy는 결과가 그래도 비슷하게 나오는 반면, DAC와의 결과는 많은 차이를 보였다.