1. **목적**

* **Class 를 디자인할 수 있다.**
* **동적 메모리 할당, 해제를 할 수 있다.**
* **Vector class를 사용할 수 있다.**
* **Knapsack Problem을 이해한다.**
* **Dynamic Programming (동적계획법) 을 이해하고 적용할 수 있다.**
* **프로그래밍을 통한 문제 해결 능력을 키운다.**

1. **문제**
   1. **문제개요**

**본 프로그래밍 숙제에서는 multiple-choice knapsack 문제의 해를 동적계획법을 활용하여 구한다. Knapsack 문제란 주어진 item들 (weight와 profit 정보를 가짐) 중 knapsack의 capacity를 초과하지 않으면서 최대의 profit을 만들 수 있는 조합을 찾아내는 문제이다. Multiple-choice Knapsack 문제는 item들이 그룹에 속하게 되고 그룹에서는 한 개 이하의 item만 선택될 수 있는 제약이 추가된다. 최적해가 여럿일 때 모든 최적해를 찾는 것은 option 사항임 (bonus 5점).**

* 1. **문제 설명**

**본 과제에서는 다음에 부분적인 정보가 주어진 class 를 작성하여야 한다. 모든 member들은 private인 것에 주의 바람. 또한 메모리 할당을 한 경우 적절하게 메모리 free를 하여야 함.**

**class Citem**

**{**

**int group\_id; // item의group\_id**

**int item\_id; // item의id number**

**int weight; // item의weight**

**int profit; // item의profit**

**// 다른 members 혹은 member functions (이후는 public 으로 선언 가능)**

**};**

**class Cknapsack**

**{**

**vector <Citem> my\_item\_list; // item 을저장한다**

**int item\_num; // 주어진item의개수**

**int capacity; // 주어진knapsack의capacity**

**int best\_objective\_value;// best objective value (찾아야할값임)**

**// 다른 members 혹은 member functions (이후는 public 으로 선언 가능)**

**};**

**본 과제에서는 다음과 같은 형식의 파일(sample\_item\_list\_10.txt)을 읽는다. 문제파일의 형식은 변하지 않지만 몇 개의 데이터가 들어 있을지는 정해져 있지 않다고 가정한다. 즉, 데이터의 개수는 동적으로 변동될 수 있다.**

**group\_no, item\_no, weight, profit**

**1,1,1,2**

**1,2,3,9**

**2,1,4,3**

**2,2,3,8**

**2,3,3,10**

**3,1,1,6**

**3,2,5,4**

**4,1,10,10**

**4,2,3, 5**

**4,3,2, 4**

* 1. **알고리즘 개발**

**본 문제를 해결하기 위한 Dynamic Programming Recurrence Equation을 개발하여 보고서에 기술하여야 한다.**

**해를 구하는 알고리즘은 다음 1)이나 2)중 하나를 응용할 수 있다.**

**1) Recursive Dynamic Programming**

**2) Iterative Dynamic Programming (Matrix base)**

**3) 최적해가 하나 이상일 때 모든 최적해를 찾는 것은 option 사항임.**

* 1. **프로그램 실행**

**프로그램은 프로그램명 input\_file capacity output\_file 인자(parameter) 3개를 받아서 구동되게 한다. Input sample data file 들은 과제 공지 파일에 함께 제공된다.**

**예를 들면, my\_m\_knapsack.exe sample\_item\_list\_10.txt 7 solution.txt 와 같다.**

**2.5 예외 처리**

**가능해가 없는 경우에는 가능해가 없다는 메시지를 화면에 출력해야 한다. 사용자 입력 오류(예, 음수 입력, 실수 입력, 개수가 맞지 않은 숫자 입력 등)는 가능한대로 예외 처리할 것을 권장하지만 필수사항은 아님.**