华南理工大学

《课程名称》课程实验报告

实验题目：The Knapsack Problem Solved by Greedy Algorithm

姓名： 谭演锋 学号： 202130100456

班级： 计算机全英联合班 组别： 无

合作者： 无

指导教师： 颜小洋

|  |
| --- |
| **实验概述** |
| 【实验目的及要求】  实验目的  Given some items with weight and profit, to finish the program to solve it using Greedy algorithm and Search tree algorithm, compare its performance.  实验要求：  The template should be used for all kinds of data type, such as: integer, real, double, etc. in the program.  Programs should be made by Object-Oriented Programming (OOP) method.  The results should be compared with ones of other algorithms, such as: Straight selection sort, insert sort, etc., and draw the graph to find their differences.  Write down the report in which there should be the execution results of the program.  【实验环境】  操作系统：Windows XP |
| **。** |
| 【实验过程】   1. 实验步骤： 2. Definition of class and function：   Struct: Goods  float weight: weight of goods  float value: value of goods  float P: value/ weight  float N: partition putting in bag range:(0,1)  function: compare: compare P between two goods.  **Class:** Knapsack  **Data member:**  bag: an array store the goods  c: capacity of the bag  n: number of goods  **Member functions:**  Knapsack: constructer for initialization the data member and sort the goods by P in descent  Greedy: use greedy algorithm to select goods putting in the bag  get\_n: return n  get\_bag: return bag  get\_C: return c  print\_bag: print the selected goods   1. Run program in the main function:   Initialize the goods. Then call the greedy method to select the goods into the bag. Then use the get\_bag function and get\_n function the print the result   1. 实验数据：   Bag capacity: 10  Goods:   |  |  |  | | --- | --- | --- | |  | Weight | Value | | 1 | 2 | 4 | | 2 | 3 | 7 | | 3 | 3 | 12 | | 4 | 1 | 6 | | 5 | 5 | 6 |  1. 实验主要过程：   Greedy:  Input: an array sorted (according the value/weight in descent) goods g1, g2,…,gn  Output: The portion of goods putting in bag x1, x2 ,…,xn  A := 0  For i:=0 to n-1 do  Begin  If A+gi.weight ≤bag \_capacity then  A+= gi.weight  xi :=1  else  xi:= (bag \_capacity – A) / gi.weight  A+= xi \* gi.weight  End   1. 实验结果： |
| **小结** |
| After writing the code for using the greedy method for solving the Knapsack Problem, I have a deeper understanding of the greedy method. The nature of the method is to choose one of the components of the solution which seems like the best one when we only see the solution of next step. It is an effective method since it cuts off multiple possible solutions which does not seem like the best at the stage of current program can know. However, since it is a greedy method, like the real world, it will be trapped in the local maximum value. However, in the Knapsack Problem, we used the greedy method, selecting the goods according to the order of the price(value/weight). In this case, we can get the global maximum since we can prove that each step, we select a thing we can get a global maximum. There are no other strategies that is better than the method we find by the greedy algorithm. |
| **指导教师评语及成绩** |
| 评语：  成绩：           指导教师签名：                                                 批阅日期： |