Department of Electrical and Computer Engineering
EECE 4542: Advanced Engineering Algorithms

Project #1

In this project you will implement exhaustive algorithms to solve two classic optimization problems. An exhaustive algorithm attempts to solve a problem using brute force to try every possible solution value.

Try to find the best solution possible for each instance using up to 10 minutes of computer time per instance.

1. Knapsack

   `void exhaustiveKnapsack(knapsack &k, int t)`

   Write a function that selects objects from knapsack instance `k` with the largest total value without exceeding the limit on the total cost. Use an exhaustive algorithm that runs for up to `t` seconds of computer time.

2. Graph coloring

   `int exhaustiveColoring(graph &g, int numColors, int t)`

   Write a function that colors nodes in graph coloring instance `g` with `numColors` colors to minimize the number of conflicting nodes. Use an exhaustive algorithm that runs for up to `t` seconds of computer time.

A collection of input files is included as part of this project. Each input file `knapsackXXX.input` or `colorXXX.input` contains an instance for a particular problem. The format file explains how data is organized within each file. Code to read instances from the files and print them out is included within the `knapsack` and `graph` classes.

Along with your code, you should submit an output file `knapsackXXX.output` or `colorXXX.output` for each instance. The knapsack output files should be produced using the `printSolution` function, and should give the total value, total cost and objects selected for the best subsets found for each knapsack instance.

The graph coloring output files should give the total number of conflicts and the color of each node for the best coloring found for each graph coloring instance. The coloring output files should be produced using the `printSolution` function, and should give the total number of conflicts, and the color assigned to each node for each instance.

For every project you complete, submit a file called resultsX.txt, where X is the project number. Within this file, write a paragraph that summarizes the performance of the algorithms you implemented.