ASSIGNMENT 1: TOWER ALLOCATION

Goal: The goal of this assignment is to take a complex new problem and formulate and solve it as (local) search. Formulation as search is an integral skill of AI that will come in handy whenever you are faced with a new problem. Local search has the characteristic that it may not find the optimal solution, but is usually able to find good solutions for really large problems. So, it is very useful in practice.

Scenario: You are the telecom minister in the government of a developing country. You want to auction off construction of cell-phone towers to various cell-phone companies. You have divided the country into M regions and have asked the companies to bid prices for each region. However, the companies are smart. They know that their profits will increase if they can capture a whole state (or a set of nearby states) instead of just a smaller region within a state. So, instead of bidding per region they decide to bid on a *set* of regions at a time. Example, a company may say, that it will pay $10,000 if it gets to build towers in regions 1, 3 and 5, however, it will pay only $3,000 if it is allowed just regions 1 and 3. Your government wants to find which bids to accept (that is give which region to which company) in order to maximize its own profits. We can assume that all companies will do a high quality job and are equal in every other way. Your government has another goal – it wants to be fair to all companies, and so you have decided to **not accept more than one bid for any company.** Note that you may choose to not accept any bid of a company. Also assume that for the purpose of first allocation that some region may go undeveloped.

Problem Statement: There are M regions to be auctioned off. There are B bids and C companies. Each bid has a company id, a set of regions, and a dollar amount. Find which subset of bids to accept so that you maximize the total auction amount. The total auction amount is sum of dollar amounts of an accepted bid. Of course, you cannot allocate one region to two companies. And, you will not accept two bids of the same company.

Algorithm 1: Implement this optimization problem as local search. Define a state representation and a neighborhood function. Implement hill climbing with random restarts as your first baseline solution.

Algorithm 2: Think about the problem further and develop your *best* algorithm for the problem. You are welcome to use any techniques discussed in the class. You are welcome to develop your own novel algorithm or your tweaks on an existing technique discussed in the class. However, you may *not* model the problem as linear programming or integer linear programming. If you are confused about whether it is kosher to use a technique, email us to ask!

Input format:

Time (in mins)

M

B

C

Company-id dollar-amount region-id region-id … region-id #

Company-id dollar-amount region-id region-id … region-id #

Here is an example

5.5

6

4

2

0 3000 0 1 4 #

0 2000 0 1 5 #

1 1000.5 2 3 #

1 1525.75 0 1 2 3 4 5 #

For this problem you are given 5.5 mins of wallclock time to solve the problem. The optimal solution is to accept bid numbers 0 and 2 ($3000 and $1000.5) so total auction cost is $4000.5. Note that the bids are implicitly numbered from 0 to B-1.

Output format:

bid-id bid-id … bid-id #

In our example

0 2 #

Code: Your code must compile and run on **machine named 'todi' or any machine with similar configuration present in GCL.** Please supply a compile.sh script for compilation. Also supply a shell script run.sh. Executing the command ./run.sh inputfile outputfile should run your code taking in the input from inputfile and outputting the solution to outputfile. We will treat both randomized and deterministic algorithms the same way, running the code multiple times till the time limit on the problem is hit, and take the best solution amongst all the runs.

What is being provided?

Your assignment packet contains 20 sample problems and sample code in C++ that (1) reads the sample problem, (2) generates a random feasible solution, (3) evaluates the random solution and (4) outputs the random solution in the desired format. The function ReadFile reads bids into a convenient data structure and GenRandom computes a random feasible solution and returns the total revenue.

What to submit?

1. Submit your code for Algorithm 2. **The code should be contained in one file named in this format -> FIRSTNAME_ENTRYNUMBER.cpp example – Nikhil_2010CS50046.cpp.** If there are two members in your team it should be called FIRSTNAME1_ENTRYNUMBER1_FIRSTNAME2_ENTRYNUMBER2.cpp
2. Submit at-most 1 page writeup (10 pt font) describing your choices and rationale for Algorithm 1 and Algorithm 2. This is not graded but failure to submit a satisfactory writeup will incur negative penalty of 20% of total score. Your writeup will help us identify any common misconceptions and particularly good ideas for discussion in the class.

Evaluation Criteria

1. Final competition on a set of similar benchmark problems. The points awarded will be your normalized performance relative to other groups in the class.
2. Extra credit may be awarded to standout performers.

What is allowed? What is not?

1. You may work in teams of two or by yourself. If you work in a team of two then make sure you mention the team details in the write-up.
2. You are required to work in C++ for fair comparison amongst all teams. You may choose to not use the sample code.
3. Your code must be your own. You are not to take guidance from any general purpose AI code or problem specific code meant to solve this or related problem.
4. It is preferable to develop your algorithm using your own efforts. However, we will not stop you from google searching.
5. You must not discuss this assignment with anyone outside the class. **Make sure you mention the names in your write-up in case you discuss with anyone from within the class outside your team.** Please read academic integrity guidelines on the course home page and follow them carefully.
6. Your submitted code will be automatically evaluated against another set of benchmark problems. You get a zero if your output is not automatically parsable.
7. We will run plagiarism detection software. Any team found guilty will be awarded a suitable penalty as per IIT rules.