CS 336: Algorithms | **Problem Set 2**
Date: Friday, October 1, 2021 | **Due:** Thursday, October 7, 2021

Submit your solution on Canvas.

Problem 1. A start-up Tech336 runs shuttles for its employees between its two campuses in Evanston and Downtown Chicago. In the beginning of the day, each employee who wants to take a shuttle from Evanston to Chicago fills out an online form, in which he or she specifies the desired time interval $[a, b]$ when he or she needs to depart from Evanston. For example, an employee can tell the system that he or she needs to take a shuttle that departs from Evanston between 3:05pm and 3:45pm. We denote the request from employee i by $[a_i, b_i]$. Shuttles can depart from Evanston only at designated times $t_1, t_2, \dots, t_m$. Given a list of intervals $[a_i, b_i]$, we need to choose a subset of departure times in $\{t_1, \dots, t_m\}$, when shuttles should depart from Evanston.

Your goal is to design an algorithm that minimizes the number of departures. The algorithm must satisfy the following constraints on the schedule:

a. Every employee who wants to take a shuttle should be given a ride.

b. Every employee should be assigned a shuttle that departs in the time interval $[a_i, b_i]$.

We assume that every shuttle can serve an unlimited number of employees.

Please, do the following:

(5 points) 1. Give a definition of a feasible solution for this problem.

(15 points) 2. Design an algorithm for this problem.

- Provide a concise pseudocode for your algorithm; and
- Describe it in words.

(5 points) 3. Show that your algorithm always returns a feasible solution if it exists.

(15 points) 4. Prove by induction that your algorithm stays ahead of the optimal solution. Specifically, show one of the following statements: *(1) every employee who takes one of the first i shuttles in the optimal solution also takes one of the first i shuttles in the algorithmic solution;* or *(2) the departure time of shuttle number i in the optimal solution is less than or equal to the departure time of shuttle i in the algorithmic solution.*

- Clearly, state the inductive hypothesis.
- Prove the base case of induction.
- Prove the inductive step.

(5 points) 5. Using item (4), explain why your algorithm finds the optimal solution.

(5 points) 6. What is the running time of your algorithm?

(5 points) 7. Explain how to modify your algorithm if every bus has a capacity of c passengers? In this case, at most c employees can take a single shuttle. However, we are allowed to schedule more than one shuttle at some departure times t_j.

Problem 2. Alice has chosen n holiday gifts for her friends. Alice knows that each gift will be on sale during Black Friday. The Black Friday sale price of gift i will be `giftPrices[i].sale`. The price of item i before Black Friday will be `giftPrices[i].before`; and the price of gift i after Black Friday is going to be `giftPrices[i].after`. Alice wants to plan when to buy each gift to minimize the total cost of all gifts. The challenge is that Alice can buy at most k gifts on Black Friday. She, however, can buy any number of gifts before and after Black Friday.

Design and implement an algorithm for finding the minimum total cost of all n gifts. Write the following function

```
MinCost(const std::vector<Price>& giftPrices, int k)
```

The parameters of this function are `const std::vector<Price>& giftPrices` – the list of prices and `k` – the maximum number of items Alice can buy on the Black Friday sale. Each element of the array/vector `giftPrices` contains three fields: `sale`, `before`, and `after`, as described above. Function `MinCost` should return the minimum possible total cost of all gifts. All numbers are integers.

Collaboration policy for Problems 1 and 2: Please, solve these problems on your own. Do not collaborate with other students.

See the next page for programming instructions.

Instructions for the programming assignment.

Download files

- `student_code_2.h` – this file should contain your solution.
- `problem_solver_2.cpp` – this is the main file in the project (don't edit this file!).
- `test_framework.h` – this is a library responsible for reading and writing data files (don't edit this file!)
- `small_problem_set_2.in` and `large_problem_set_2.in` – these files contain test problems for your algorithm (don't edit these files!)

Place all files in a new folder/directory. Write your code in function `MinCost`. Also, write your name in the function `GetStudentName`. Both functions are located in file `student_code_2.h`. Compile and run your code. To compile your code do the following.

- If you use Clang compiler, type
  `clang++ -std=c++17 -pedantic-errors problem_solver_2.cpp -O2 -o problem_solver_2`
- If you use GNU C++ compiler, type
  `g++ -std=c++17 -pedantic-errors problem_solver_2.cpp -O2 -o problem_solver_2`
- If you use Microsoft Visual C++ compiler, start `Developer Command Prompt` and type
  `cl /EHsc problem_solver_2.cpp`

Your compiler should be compatible with `C++17`. If you work in the Wilkinson Lab, you need to start developer tools first: Type

- `scl enable devtoolset-4 bash`

Once you compile your code, start your program. Type `./problem_solver_2 small` to run your code on simple problems and `./problem_solver_2 large` to run your code on hard problems. On Windows, type `problem_solver_2.exe small` and `problem_solver_2.exe large`, respectively. Make sure that executable is located in the same folder as files `small_problem_set_2.in` and `large_problem_set_2.in`. If your code works correctly, you will get the following message:

> Problem set 2. Your algorithm solved all test problems correctly. Congratulations! `solution_2.dat` via Canvas.

If your code makes a mistake, you may get a message like this:

```
Problem set 2.  Mistake in problem #15.  Correct answer:  4.  Your answer:  12.
```

Please, test your code with the both problem sets (small and large). When your code is ready, submit file `student_code_2.h` on Canvas. Make sure that you are submitting the latest versions!