CS 336: Algorithms **Problem Set 6**
Date: Saturday, November 13, 2021 **Due:** Sunday, November 21, 2021

Submit your solution on Canvas.

Problem 1. You are organizing a large event in Evanston. You are expecting many guests and you need to purchase n disposable plates to serve the food. Disposable plates are available in k different packages $\#1, \#2, \ldots, \#k$. Package $\#i$ costs p_i dollars and contains n_i plates. You need to design an algorithm that chooses which packages to order to minimize the total purchase price. Note that (1) you can order more than n plates; (2) the price per plate can be different for different packages.

Example. You need to order 1200 plates. The available packages are (#1) 500 plates for \$50, (#2) 300 plates for \$40 and (#3) 100 plates for \$30. The optimal solution to this problem is to order two #1 packages and one #2 package. In this case, you are ordering $2 \times 500 + 1 \times 300 = 1300$ plates and paying $2 \times \$50 + 1 \times \$40 = \$140$.

Your goal is to design a dynamic programming algorithm for this problem.

I. Define a subproblem for your dynamic program (DP). Make sure that your definition is concise and unambiguous.

II. Describe the base case for your DP.

II. Write the recurrence relation.

IV. Explain why the recurrence relation is correct.

V. Show how to solve the original problem using the solution to the DP.

Problem 2. You work for a startup that develops a new smartphone – Phone336. This phone must be capable of working 336 hours (2 weeks) without charge. To make it happen, you need to develop a new energy-aware job scheduling software. Your scheduler receives n jobs that need to be executed on the phone one after another. The order of jobs is given to the scheduler in advance and cannot be changed. All jobs are “parallelizable”. That is, every job can run on several processor cores at the same time. The phone’s processor has 8 cores. Thus, your software should assign every job j up to 8 cores. If job j uses k cores, then its processing time is p_{jk}, and it consumes q_{jk} units of energy.

We ask you to design an algorithm that given n jobs, numbers p_{jk}, q_{jk}, and an energy budget Q, finds the minimum time required to execute these jobs subject to the constraint that these jobs can consume at most Q units of energy. Parameters p_{jk}, q_{jk}, and Q are positive integers. The running time of your algorithm must be polynomial in n and Q.

Example. Here is an example for 5 jobs and 3 cores.

Processing times p_{jk}.

	job 1	job 2	job 3	job 4	job 5
1 core	12	10	15	12	10
2 cores	6	5	10	6	10
3 cores	4	5	8	4	10

Energy consumption q_{jk}.

	job 1	job 2	job 3	job 4	job 5
1 core	1	1	3	1	2
2 cores	1	4	9	2	4
3 cores	1	9	16	5	6

If $Q = 8$, then you can assign 3 cores to the first job and 1 core to other jobs. In this case, the total energy consumption is

$$q_{13} + q_{21} + q_{31} + q_{41} + q_{51} = 1 + 1 + 3 + 1 + 2 = 8,$$

and the processing time is

$$p_{13} + p_{21} + p_{31} + p_{41} + p_{51} = 4 + 10 + 15 + 12 + 10 = 51.$$

If $Q = 10$, then you can assign 3 cores to job 1; 2 cores to job 4; and 1 core to jobs 2, 3, 5. In this case, the total energy consumption is

$$q_{13} + q_{21} + q_{31} + q_{42} + q_{51} = 1 + 1 + 3 + 2 + 2 = 9,$$

and the processing time is

$$p_{13} + p_{21} + p_{31} + p_{42} + p_{51} = 4 + 10 + 15 + 6 + 10 = 45.$$

Write the following function

```
MinProcessingTime(const std::vector<std::vector<Profile>>& profiles, int maxEnergy)
```

The parameters of this function are job profiles (`profiles`) and energy budget (`maxEnergy`). The processing time of job j on k cores equals `profiles[j][k-1].time`. The energy consumption of job j on k cores equals `profiles[j][k-1].energy`. Your function should return the minimum running time. If there is no feasible solution to the problem, your function should return `-1`. The running time of your program should not exceed several seconds (please, compile your code with `-O2` or `-O3` optimization switch). You can assume that the maximum number of jobs in 150 and `maxEnergy` ≤ 10000.

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_6.h` – this file should contain your solution.
- `problem_solver_6.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_6.in` and `large_problem_set_6.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 `MinProcessingTime`. Also, write your name in the function `GetStudentName`. Both functions are located in file `student_code_6.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_6.cpp -O3 -o problem_solver_6
  ```
- If you use GNU C++ compiler, type
  ```
  g++ -std=c++17 -pedantic-errors problem_solver_6.cpp -O3 -o problem_solver_6
  ```
- If you use Microsoft Visual C++ compiler, start `Developer Command Prompt` and type
  ```
  cl /EHsc /O2 problem_solver_6.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_6 small` to run your code on simple problems and `./problem_solver_6 large` to run your code on hard problems. On Windows, type `problem_solver_6.exe small` and `problem_solver_6.exe large`, respectively. Make sure that executable is located in the same folder as files `small_problem_set_6.in` and `large_problem_set_6.in`. If your code works correctly, you will get the following message:

> Problem set 6. Your algorithm solved all test problems correctly. Congratulations! `solution_6.dat` via Canvas.

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

```
Problem set 6.  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_6.h` on Canvas. Make sure that you are submitting the latest versions!