1 Problem Statement

Emergency medical supplies are needed ASAP in a remote area of the globe. You are tasked with packing the supplies for shipment. The courier has provided you with a list of \( c \) different containers they can ship. Each container, \( i \in 1..c \) has a fixed capacity \( C_i \). Your inventory indicates that you have \( n \) items. Each item \( i \in 1..n \) has a size \( S_i \) and a type \( T_i \). The size value indicates how much room the item will take in a container and the type value indicates the kind of medical supplies. Due to the obscurity of the destination there is a high risk of container damage and supply contamination. Because of this heightened risk, the aid organizations have requested that no more than two types of supplies be packaged into one container. Your goal is to pack the supplies into the containers to minimize the empty space in the containers, while obeying the packing type constraint. Let \( B_j \) be the set of items you put into container \( j \), and let \( SB_j = \sum_{k \in B_j} S_k \), that is the size of all the items packed into \( j \). Then the objective is formulated as, minimize

\[
\sum_j \min_{C_i \geq SB_j} (C_i - SB_j)
\]

As this objective implies, you can use as many containers of a given size that you like.

2 Assignment

Write a algorithm to solve the Special Ops Problem. You can apply any technique you want, including but not limited to LS, CP, IP, LP, DP, brute force etc. Your algorithm should be able to perform on all of the data sets in the course directory. The assignment should be performed by teams of at most 2 people. We always expect

- both source files and binary programs, if any, of the working algorithm;
- a specification on how to compile and run the program;
- a brief report in plain text containing, the names of each team member, a brief discussion of your solution strategy, implementation techniques and experimental observations. The report should be concise. An example report can be found here,
3 I/O Specification

Format The input file contains $n + 3$ lines, where $n$ is the number of items in your inventory. The first line provides the container information. The first value is the number of container types, $c$, followed by the size of each container $C_i$. The second line is the number of item types in the inventory. The third line is the number of items in the inventory $n$. It is followed by $n$ additional lines specifying the inventory. Each inventory line, $i$, has two values, the size of an item $S_i$ and the type of the time, $T_i$. All the values are integers.

[Input Format]
|C| $C_0$ $C_1$ ... $C_{c-1}$
|T|
|N|
$S_0$ $T_0$
$S_1$ $T_1$
...
$S_{n-1}$ $T_{n-1}$

The output file contains 2 lines. The first line has two integers $l$ and $opt$ flag. $l$ is the sum of unused space in each container. $opt$ flag = 1 if the algorithm can prove the optimality and 0 otherwise. The second line is a list of $n$ values where the $i$th value indicates the container identifier that the $i$th item is put in.

[Output Format]
$l$ opt_flag
$b_0$ $b_1$ $b_2$ ... $b_{n-1}$

Example For an example input see
/course/cs258/data/special/sp_001
For an example output see
/course/cs258/data/special/sp_001.out

Instructions We will run your submission using the command: ./sp <timelimit> <filename>

For example: ./sp 300 /course/cs258/data/special/sp_001
means the program will use sp_001 as input and will run at most 300 seconds.

We use stdout for output. Output to other stream will be ignored (you may want to send runtime information to stderr). Your submission will be tested on a department linux machine. If your algorithm is a stand-alone program, please name it sp, otherwise, please specify the compilation procedure, it is appreciated if you also provide a script that follows the above format to run the program.

---

1 Item type identifiers are 1-based, not 0-based.
Resources  You can find special ops instances in `course/cs258/data/special` and an example output file, `course/cs258/data/special/sp_001.out`

4 Remarks

Handin  Command: `course/cs258/bin/cs258_handin hw7`
All of the files in the current directory and sub-directories will be submitted. Only the last submission will be marked.

Questions  Please contact the class GTA Carleton (cjc@cs.brown.edu).

Warning  This is a first time assignment, be prepared for clarifications and bug-fixes.