1 When I was your age...

Remember when we were kids, we played with marbles?\footnote{No? Just me? Maybe I’m too old ...} We used to also collect them and exchange them for different types. Clearly each type had a different value, you could trade two of these “ugly” ones for this pretty one.

Suppose you have a group of $n$ friends. Each one of them collects one specific type of marbles. Suppose also that for each pair of friends $i \neq j$, you’ve computed a ratio $r_{ij}$: one marble from $i$ gives you $r_{ij}$ marbles from $j$. Notice that $r_{ij}$ can be a fraction; that is if $r_{ij} = \frac{1}{2}$ then if you buy two marbles from friend $i$, you can exchange them for one marble from friend $j$.

A marbling cycle is a fixed order $x_1, x_2, ..., x_k$ where you successively exchange marbles from friend $x_i$ with marbles from friend $x_{i+1}$ for all $1 \leq i < k$, and you exchange marbles from $x_k$ with marbles from $x_1$. More precisely, you buy your favourite type of marbles from the friend who collects them (that’s $x_1$). You then perform a sequence of exchanges from $x_1$ to $x_2$, $x_2$ to $x_3$ etc until $x_k$ to $x_1$. And you hope that by the end of your marbling cycle, you end up with more marbles of your favourite type!

This only happens if the product of the ratios along this marbling cycle is above 1! Give a poly-time algorithm that checks whether you will be able to end up with more marbles at the end of your cycle.

2 Sucks to be broke!

Summer is still here but you’re broke already and are trying to save money for your next trip! You found a night job at a hospital: Saint Hell’s Hospital (SHH). SHH operates two main clinics, one in Toronto (SHHT) and one in Ottawa (SHHO), and by law they have to always have $k$ night guards in total between the two locations. They hired you to meet the law requirements, which makes your job more flexible (or not). For the $n$ coming months, you have to work at either SHHT or SHHO and you have the freedom to chose the clinic you go to.
For every month $i$, you make $T_i$ or $O_i$ dollars depending where you work. However, if you decide to switch clinics from month $i$ to month $i+1$, you incur moving expenses, a fixed price $E$ for the move. For simplicity, suppose all your other expenses (rent, food...), except the move, are covered by some generous God. By the end of the $n^{th}$ month, you want to maximize your savings.

**Input**: A list of values $T_1, T_2, ..., T_n, O_1, O_2, ..., O_n$ where $T_i, O_i$ is your income on month $i$ at SHHT, SHHO respectively, and a value $E$ representing the moving expense.

**Output**: A sequence $S$ of $n$ locations that maximizes $v(S)$, where location $i$ represents the clinic you will work at during month $i$ and $v(S)$ is the total value of your savings.

Devise an algorithm to solve this problem efficiently. Give a high-level description of how your algorithm works, state the time complexity of your algorithm, and prove that your algorithm is optimal.

### 3 Piw Piw!

Working for SHH isn’t so bad after all. It sure is quiet and boring at times and not much is happening on Reddit over night, but you manage to keep yourself busy. On a boring night however, you get a call from the emergency room. There was an outbreak and they need to administer the antidote to the affected patients. You were asked to come help.

You run upstairs where you are assigned a room, an antidote gun, and a list $L = \{x_1, x_2, ..., x_n\}$ of number of patients per minute that you will get to hopefully save. This means, at minute $i$, you will have $x_i$ infected patients and you want to save as many of them as possible. The gun you’re given pumps the antidote from a heated container, and the whole process works as follows:

- If you let the antidote container heat for $j$ minutes, then you get $f(j)$ dosages (shots) to cure $f(j)$ patients.
- If you use the gun at the $i^{th}$ minute, and it has been $j$ minutes since you last used it, then you will cure $\min(x_i, f(j))$ patients. Of course, if you use $\min(x_i, f(j))$ dosages, then the gun will be completely empty and you will need to wait for more dosages to be available.
- When you got to the room, the gun was empty. This means, if you started using it at the $j^{th}$ minute, then you will be able to cure up to $f(j)$ patients.

Given the list $L = \{x_1, x_2, ..., x_n\}$ and the function $f(\cdot)$, you want to select when you will use the antidote gun in order to cure as many patients as possible.

Construct an efficient algorithm that returns the maximum number of patients you can cure. Give a high-level description of how your algorithm works, state the time complexity of your algorithm, and prove that your algorithm is optimal.

### 4 Everybody deserves good health care!

You saved the most patients out of all the staff! Unfortunately many patients passed away. There just weren’t enough staff to help all the patients; so you decided to run for mayor elections again to change this situation. Everybody deserves good health services! This was your selling point at the town elections.

---

2I don’t know why they believe a night guard is capable of doing this...

3Not bad for a night guard 😂
You’ve been elected (Gee you’re good at politics) and as a good politician *cough again* you want to improve everyone’s access to hospitals and health services in general.

A good access to health services means two things: 1. A household should not travel further than $s$ kilometers to get to a hospital, and 2. no hospital should be responsible for more than $t$ households.

Before investing into building new hospitals, you want to check if the current setup of the town is good\(^4\) (i.e. it meets the requirements above).

**Input**: A list $A$ of $n$ houses and a list $B$ of $m$ hospitals, where every house and hospital is represented by a pair $(x, y)$ on the place, and $s$ and $t$.

**Output**: “Good” if each household has access to a hospital under the restrictions above, and “Eeek” otherwise.

Devise an algorithm to solve this problem. Give a high-level description, state the time complexity of your algorithm, and prove that it is optimal.

5 \hspace{1cm} \textbf{Eeek!}

It turned out the town needs one more hospital. You selected the location and hired all the necessary staff, $n$ people in total $p_1, p_2, \ldots, p_n$. There are different tasks ($k$ of them) to be done at the hospital, from surgery to mopping the floor etc. Suppose there are $n$ tasks to be completed in a given day, and assume (clearly) that $n \geq k$ (might have to perform more than one surgery in a given day). Suppose that each task takes one hour to complete, and a task can only be completed by one person. Each person $p_i$ works a total of $h_i$ hours per day and can perform a subset $S_i$ of the of task types.

Give an algorithm that checks whether it is possible to get all the tasks done in a given day. If it is possible, show how to extract the list of jobs each $p_i$ will have to complete. Prove the correctness of your algorithm.

---

\(^{4}\)the breakout night was an exception..