VE477
Introduction to Algorithms

Challenge

Manuel — UM-JI (Fall 2020)

- Abstract a real life problem
- Find an algorithm to solve a problem
- Prove its correctness and complexity
- Rewarded by a bonus on the final grade

Although the Fall semester is pretty busy you decide to take an internship in order to improve the quality of your graduate school applications. After a bit of search and a few unsuccessful interviews you eventually resolve to accept the offer from TiiJ, an IT company with a web-development department.

On your first day, you enter the office and wait to be assigned some tasks. Quickly you are explained that you will join the web development team. This is a great news as you have already designed a few websites with HTML5 and CSS. You are also very respectful of the web standards and familiar with tools such as `https://validator.w3.org/`, as in fact you proudly exhibit all the validation icons on the websites you have designed.

During the next few hours you are explained in details all the requirements you have to match. Among the most emphasized points you learn that you must (i) use the last version of Microsoft Front Page Express to write the websites; (ii) include as many buttons as possible (even if one is enough); (iii) when a hidden box is expanded, do so as high as possible above the button which opened it and do not notify the user; (iv) as much as possible do not disable or hide irrelevant information, simply include it the middle of the useful content; (v) use and abuse pop ups; (vi) feel free to include Chinese text in the middle of an English text; (vii) if a page includes videos, ensure they are all fully downloaded before the user can do anything; You also learn that they pay much attention to the quality of their product, and as such you should never forget to test your website, IE 6 being recommended.

As you have never heard of Microsoft Front Page Express you hope it is a new product which like VSCode, is available on all platform. Indeed you are working on a new Linux project running heavy computations in the background and cannot afford to stop it for too long too often. Unfortunately you quickly discover Front Page express was last release in 2003! When you ask if you could use another tool available on Linux you are first asked what is Linux, and then clearly explained that for obvious quality and consistency reasons you should stick to the TiiJ guidelines. Then you ask them if you could have another computer but they explain that due to budgetary constraints they cannot give a computer to each intern. Therefore you have to use your one and only computer.

As you have no idea on how you could solve this dilemma, you try to recall everything you learnt in ve477 on how to tackle new problems. After some thinking you come up with the following strategy.

- List all the T independent computation tasks to run on Linux and determine the number of minutes each of them will takes; you denote the length of the ith task by t_i.
- Check what are the P webpages that you should write as well as their expected completion date and the evaluation of the time it will take you to do it; Being sure of your time estimation e_i for page p_i you decide to always start your work at the last minute, meaning that if a page p_i has to be completed at time t_i and you will start it at time $s_i = t_i - e_i$. In particular it appears that sometimes you can have more than one page to work on at a time.
- After discussing with a friend he accepts to help you with no more than two webpages if you

complete his homework. On the other hand you want to be careful and not be caught by HC so you do not want to complete more than H homework exercises for him. He has a quick look at the list of pages to implement and rates each of them in term of exercises, i.e. he can write page p_i for you if you complete h_i exercises for him.

Now that you have expressed all the parameters you want to calculate the earliest time each computation task can complete while respecting all the above constraints.

Rules

Follow the rules below:

- Prepare a proof of correctness and determine the complexity of the proposed algorithm;
- Implement the proposed algorithm in OCaml or Python;
- Test the correctness of the solution on JOJ;

Reward

The reward will de determined as follows.

- Single: up to 3 extra marks on the final course grade;
- Team: up to 6 extra marks on the final course grade to be equally shared among the team members;
- The first to solve the problem will get a full reward;
- The second to solve the problem will get half of the reward;
- The third to solve the problem will get a quarter of the reward;
- Extra bonus will be awarded for submission written in OCaml;