• describe your greedy method and
• give a proof of correctness of your method (using an exchange argument).

You are running a software company and have a series of $n$ jobs that must be pre-processed first on a supercomputer before being moved to a smaller PC. You have only one super-computer, but you have $n$ PCs so the second stage can be performed in parallel. More specifically, your jobs are described as $J_1 = (s_1, f_1), J_2 = (s_2, f_2), \ldots, J_n = (s_n, f_n)$, where job $J_i$ needs $s_i$ units of time to be pre-processed on the super-computer and $f_i$ units of time on the PC.

You need to work out an order in which to give the jobs to the super-computer. As soon as the first job is done on the super-computer, it can be moved to the PC for finishing; at that point a second job can be given to the super-computer; when the second job is done it can go straight to a PC since the PCs can work in parallel, and so on. So if the jobs are processed in the order given, job $J_i$ finishes at time $\sum_{k=1}^i s_k + f_i$.

A schedule is an ordering of the jobs to be given to the super-computer. The completion time is the point at which all jobs have finished being processed on the PCs. We wish to minimize the completion time.