

Problem Set 2 TA comments

There are several common problems for 4.7:

1. Not explain why the completion times of all jobs except the ones being swapped stay the same. (So that we do not need to consider those)
2. Swap two jobs J_l and J_k that are not neighboring. In this case, all the jobs between J_l and J_k also change their finishing times.
3. Just give an intuition of how to design the algorithm without a formal proof.

There are many different kinds of problems for 4.14. The basic idea is to use exchange argument. Many students forget to prove at least one or two of three steps in the solution. (Please refer to the solution.)

- a. Why s_j cannot be bigger than t_j ?
- b. What if s_j is smaller than t_j ?
- c. Show that $k = m$

Another way given by some students is to use the interval scheduling as a black box. (But in this case you won't have chance to practise exchange argument, :-)) Our algorithm for this problem is exactly the one used for interval scheduling. So the number of `status_check` by our algorithm equals to the size of the largest set of disjoint processes. (We already know interval scheduling will find the largest set of disjoint processes.) The number of `status_check` by our algorithm must be not greater than those by any other algorithms. (we need at least one `status_check` for each disjoint processes).