

These comments are no replacement for coming to talk to me in office hours. I am here to help!

Comments on Homework 4 and Quiz 4

How do we solve the machine-scheduling problem / apply the list processing algorithm? It comes in two parts:

- an order-requirement digraph. This tells me what is physically possible at each moment in time. A task is not ready until all of the tasks before it (on the order-requirement digraph) are completed. If T_1 must be performed before T_2 , then it is not enough to assign T_1 before T_2 . I must wait T_1 is finished, before I can assign T_2 .
- a priority list. This tells me what order I would prefer to do things in (if I did not have to worry about the order-requirement digraph). Suppose my priority list starts $T_3, T_4, T_1, . . .$. I start by looking at the order-requirement digraph and asking "Is T_3 ready?" If it is ready, then I assign it to the first free processor. If T_3 is not ready, then I move onto the next item on the list and ask "Is T_4 ready?" I continue in this way until I find a ready task. The only time a processor idles is if NO tasks are ready. In particular, the priority list does NOT mean "I must do T_3 before I do T_4 ". It means I ask "Is T_3 ready?" before I ask "Is T_4 ready?"
- If T_3 was not ready and T_4 was ready, I would assign T_4 to the first free processor and cross T_4 off the priority list. When I next try to assign a task, I start at the beginning of my priority list $T_3, T_1, . . .$, with T_3 and not T_1 .

It is important to know what tasks are ready. In class, I have suggested two things which may help.

- Keep track of how much time has passed.
- Cross tasks off the order-requirement digraph only when they have been *finished*, not when they are assigned.

About Bin-Packing

- Sadly we did not talk about NFD and FFD on Wednesday. Without parts b) and d), Problem 46 was not a good example of the difference between NF and FF. I apologize for that.
- It is a good idea to keep track of the total amount stored in each bin. If I put weights of size 3 and then 4 into a bin, I should write 7 to the side of the bin. This will help prevent me from overfilling bins and help me identify what bin I should use for the next object.