很明显，超级计算机的工作时间不取决于任务的排序。因此，当最后一个任务移交给 PC 时，我们无法改变时间。直观上很清楚，调度中的最后一个任务应该是完成时间最短的任务。

这种非正式的推理表明以下贪心调度应该是最优的。

调度 G：按完成时间递减的顺序运行任务 f_j。

现在我们将使用**交换论证 (Exchange Argument)** 来证明 G 实际上是最优调度。我们将证明，对于任何给定的调度 $S \neq G$，我们可以重复交换任务，将 S 转换为 G，而不会增加完成时间。

考虑任何调度 S 且不使用 G 的顺序。那么这个调度必然包含两个任务 J_k 和 J_l，使得 J_l 在 J_k 之后运行，但是第一个任务的完成时间小于第二个任务的完成时间，即 $f_k < f_l$。我们可以通过交换这两个任务的顺序来优化调度 S。设 S' 是我们交换 J_k 和 J_l 顺序后的调度。很明显，除了 J_k 和 J_l 之外，所有任务的完成时间都没有改变。任务 J_l 现在调度提前了，因此这个任务在新调度中的完成时间将早于原调度。任务 J_k 调度滞后了，但是超级计算机在新调度 S' 中将 J_k 移交给 PC 的时间与它在原调度 S 中移交给 J_l 的时间相同。由于 J_k 的完成时间小于 J_l 的完成时间，任务 J_k 在新调度中将比 J_l 在原调度中更早完成。因此，我们交换后的调度并没有更大的完成时间。

如果我们在 S 中定义一个逆序（如文本中所述，一对任务的顺序与其完成时间顺序不一致），那么这样的交换会减少 S 中逆序的数量，而不会增加完成时间。通过一系列这样的交换，我们可以将 S 转换为 G 而不增加完成时间。因此，G 的完成时间不大于任意调度 S 的完成时间。所以 G 是最优的。

交换论证的注意事项

与所有交换论证一样，存在一些需要注意的常见错误。我们在这里总结其中一些；它们也适用于其他问题。

- 交换论证应该从一个任意调度 S 开始（特别是，它可以是一个最优调度），并使用交换来证明这个调度 S 可以被转换为算法 G 产生的调度，而不会使总完成时间变差。从算法调度 G 开始并简单地论证 G 不能通过交换两个任务来改进是行不通的。这个论证只会表明从 G 通过一次单一交换获得的调度并不更好；它不会排除通过多次交换可以获得更好的其他调度的可能性。
- 为了使论证顺利进行，应该交换**相邻**的任务。如果你交换两个不相邻的任务 J_l 和 J_k，那么介于这两个任务之间的所有任务的完成时间都会改变。
- 通常，以上交换论证不能以**反证法**来表述——也就是说，考虑一个最优调度 O，假设它不等于 G，并得出矛盾。问题在于可能存在多个最优调度，因此存在一个不等于 G 的最优调度并不矛盾。请注意，当我们交换相邻的逆序任务时，这不一定会使调度变得更好；我们只论证这样的交换不会使其变得更糟。

最优性的替代证明

最后，值得注意的是，调度 G 最优性的以下替代证明不直接使用交换论证。设 J_j 是贪心算法调度 G 中在 PC 上最后完成的任务。设 S_j 是这个任务在超级计算机上的完成时间。那么总完成时间是 $S_j + f_j$。在任何其他调度中，前 j 个任务中的第一个，在 G 中指定的其他顺序下，必须在超级计算机上在某个时间 $T \geq S_j$ 完成（因为前 j 个任务正好将 S_j 的工作量交给超级计算机）。现在设这个任务是 J_k。现在任务 J_k 需要的 PC 时间至少与任务 J_j 相同（由于 G 的排序），所以它在时间 $T + f_k \geq S_j + f_j$ 完成。说明这个其他调度并不更好。