- Algorithm with Gurobi for case 1: very fast, can take external order other than files
- Algorithm with Gurobi for case 2: too slow
- Analitic algorithm, greedy approach (given order)
- Gurobi #2 doesn't find easy improvements, can't finish the search and ignores same-order different OR(i) cases, easily explorable by gurobi #1
- Greedy algorithm not optimal even for 3 jobs sometimes
- Greedy algorithm going backward, not optimal as well
- Greedy modified algorithm, better but still problems (requires knowledge of many patients before)
- Ultimate Gurobi 2: full execution in ≃ 4.5 min for 15 irrelistic patients, much simpler
  ≃ 25.6 min for 16 irrelistic patients
  ≃ 2.6 min for 14 irrelistic patients
- Algorithm to create a schedule with imposed placements.
- Created set of optimal solutions.
- B&B WIDTH_2 ($C^I < C^{II}$): final solution is not optimal, i have good reason to believe is equal to greedy forward.
  The two branches end up being the same branch as a single job of difference never creates an immediate advantage, and when it could have created one the decision has already been taken.
- B&B, unbound width, $C^I < C^{II}$ AND $GAP^I > GAP^{II}$: The algorithm is fairly slow but maybe due to implementation method more than complexity itself.
  While the optimal solution is found, around 2/3 of the original tree is kept at last stage when comparison occurres just between nodes of the same node.
  The optimal solution doesn't seem to be unique, but it may be due to branches where the choice is irrelevant.
  Interesting: since i am comparing branches of same node, one of the two never has a gap.

- B&B, pruning decision taken by separating at each level OR=1 from OR=∅. Then confronting all the nodes of each group respectively with:
  OR=1 → $OF^I \leq OF^{II}$ AND $C.AOP^I \leq C.AOP^{II}$ ⟹ Prune solution II
  OR=∅ → $OF^I \leq OF^{II}$ AND $GAP^I \geq GAP^{II}$ ⟹ Prune solution I
  Results: always finds optimal solution, instantaneous runtime and at most 6 nodes per level.

OPTIMAL SOLUTION FOR PROBLEM 1

- Found greedy for case 2, not optimal, the more patients the better. It finds very good sequences but breaks final part very sketchy.
- Made algorithm for swapping two patients. Also made version that researches whole swap-space 0 1 2 3 4 ⟶ ∅ 4 2 3 1
- Made same thing for sliding patients ∅ 1 2 3 4 5 ⟶ ∅ 1 5 2 3 4
- Created function to slide around group of patients.

- Found a solution better than Gurobi:
  - solution is not incorrect
  - solution doesn't break any constraint of Gurobi model, in fact forcing the solution
  - What do i do?