1. Implementera datastrukturen
   1. Indata
      1. Pick n’ leave matriser
      2. toolTravelTime-matriser
      3. toolTimeWindowMatriser
   2. nod-class (senare)
      1. Förslag på innehåll:
         1. Component
         2. Fixture (1-2)
         3. \*
         4. Component (1-5) + 2 (peel))+1(final assembly)
         5. Tray (1-5)
         6. \*
         7. Camera (1-5)
         8. \*
         9. Air flow
         10. Output
         11. Is start node
         12. Is end nod
      2. Component class
         1. Suction
         2. Photo (bool)
         3. grip
2. Bivillkor modellen
   1. Paths: Next[i] == j ⬄ Prev[j] == I (**Jocke**)
      1. Kolla så det funkar för start & end nodes
   2. Routes: Route[i] == Route[Prev[i]] (Ludde)
      1. Route[i] == Route[Next[i]]
      2. Dubbelkolla, och få funka explicit för start- & end nodes
   3. NoCycles måste funka (inge cykler) (**Jocke**)
   4. Cumulative för tid, gripper & suction ska funka
      1. Callback för tid
      2. Indata (-1, 0, 1)-värda matriser för grip & suction pickups
         1. Ändrade till callback som returnerar [-1,1]
      3. Någon vettig indata för pickup callbacks
   5. Verifiera att callback funkar (Joakim)
      1. Gör och kontrollera 2.5D callback (active, prev, route)
   6. Tilldela enbart en route per route (Route[i] = Route[Next[i]) Kolla – funkar inte för end nodes & start nodes
3. Implementera bi-modellen” (efter node-struktur)
   1. Se till att ”inaktiva” noder effektivt hamnar i route 0
      1. Denna ska tilldelas fullständigt när route 1 & 2 är tilldelade.
   2. GCC på tray- & camera-noder för tilldelning till route 0
   3. Skalärprodukt ( & inequality ) på fixturnoderna
      1. Ändrat till multiplikation == 0 och i != i+5
   4. Slutmontage är i början/slutet av rutten
   5. Precedence baserat på input (Timewindow)
4. Read-metoder
   1. XML-input-format
   2. Strukturera input objects.