Problem 1: Geometrical Representation

Consider the following data structure used to represent a set of rectangles on a single plane. Rectangles on the layer are sorted first by their lower-left X coordinate value, with the smallest value first in the list. If the multiple rectangles have the same value of $X_{LL}$, they are sorted in a sub-list by $Y_{LL}$, with the smallest value in the list.

\[ X_{LL}: \quad \text{smaller} \quad \rightarrow \quad \text{larger} \]

\[ Y_{LL}: \]

\[ \begin{array}{c}
X_{LL} \\
Y_{LL} \\
X_{UR} \\
Y_{UR} \\
\ldots \\
\Phi
\end{array} \quad \begin{array}{c}
X_{LL} \\
Y_{LL} \\
X_{UR} \\
Y_{UR} \\
\ldots \\
\Phi
\end{array} \quad \begin{array}{c}
X_{LL} \\
Y_{LL} \\
X_{UR} \\
Y_{UR} \\
\ldots \\
\Phi
\end{array} \]

\[ X_{LL}: \quad \text{larger} \quad \rightarrow \quad \text{smaller} \]

\[ \begin{array}{c}
X_{LL} \\
Y_{LL} \\
X_{UR} \\
Y_{UR} \\
\ldots \\
\Phi
\end{array} \]

a) Write an algorithm in pseudo-code to insert a rectangle $R_A$, in the data structure.

b) Write an algorithm in pseudo-code for finding and returning all pair-wise combinations of rectangles that intersect one another. (abutment is not intersection)

c) Write an algorithm in pseudo-code for finding and returning all rectangles that are fully enclosed by another rectangle.
Problem 2: Static Timing Analysis

a) Given the circuit below labeled with arrival times, wire delay, and gate delays, compute the critical path(s) and highlight them. Formulate the problem as a graph search problem, you can assume the circuit has only one primary output. Sketch the algorithm you will use for the problem. For each gate give the largest path delay to the output of that gate by running your algorithm.

Primary input arrival time: (a, 1), (b, 2), (c, 0.5), (d, 0.7), (e, 1.5), (f, 3), (g, 3.5), (h, 3.7), (i, 6.0)
Gate delay: (G1, 4), (G2, 2), (G3, 2), (G4, 3), (G5, 3), (G6, 1.5), (G7, 1.5), (G8, 2), (G9, 2), (G10, 2), (G11, 1.5), (G12, 2), (G13, 2), (G14, 2)
Inter-gate delay: (G1, G2, 3), (G1, G5, 4), (G2, G3, 5), (G3, G8, 2), (G3, G7, 3), (G3, G11, 3.5), (G4, G3, 1.5), (G5, G12, 3.7), (G6, G8, 2.1), (G7, G9, 1.2), (G8, G10, 1.7), (G9, G10, 2.3), (G10, G13, 1.8), (G11, G12, 1.9), (G12, G14, 2.7), (G13, G14, 2.1)

b) What is the longest path in the circuit below? What’s the true delay of the circuit? Mux delay is 5. The rectangles are delay elements, labeled with their delay. The arrival time for the primary inputs is zero.
Problem 3: Compaction

Given the following constraints:

\[ a = 0, b = a + 2, c = b + 1, d = b + 2, \ m \geq a + 1, n = m + 1, o = m + 2, e \geq a + 2, f = e + 1, g = f + 1, \]
\[ h = f + 2, p \geq m + 3, q = p + 1, r = p + 5, s \geq p + 2, s \geq e + 6, t = s + 2, u = s + 2, i \geq s + 2, \]
\[ i \geq e + 3, j = i + 3, k = j + 1, y = j + 2, z \geq i + 4 \]

a) Derive the forest of equality trees.
b) Derive inequality constraint graph.
c) Show relationship between the two, solve the constraint graph and give optimum values for each vertex.
d) What’s the complexity of solving equality/inequality constraints?

Problem 4: Partitioning

Apply the Kerninghan-Lin algorithm to the graph shown below, where the areas associated with each vertex are:

\[ V1 = 10, V2 = 12, V3 = 8, V4 = 15, V5 = 13, \]
\[ V6 = 20, V7 = 9, V8 = 7, V9 = 14, V10 = 9 \]

Consider the edge weights to be equal. The areas of the two partitions should be as equal as possible. Show all intermediate steps and decisions.
Problem 5: Partitions for Timing

For the graph below, if the delay crossing the partition boundary is 20ns (e.g. from chip to chip), and the delays from the inputs to the outputs of a vertex are as listed below.

\[
\begin{align*}
d(V1) &= 3 \text{ ns} \\
d(V2) &= 2 \text{ ns} \\
d(V3) &= 1 \text{ ns} \\
d(V4) &= 2 \text{ ns} \\
d(V5) &= 3 \text{ ns} \\
d(V6) &= 4 \text{ ns} \\
d(V7) &= 3 \text{ ns} \\
d(V8) &= 8 \text{ ns} \\
d(V9) &= 7 \text{ ns} \\
d(V10) &= 5 \text{ ns}
\end{align*}
\]

a) Find the best single partition of the graph such that the longest delay between the input node (V1) and the output node (V8) is minimized and the partitions have the same size (within +/- 20%). Use the vertex size from the problem above.
b) Show the pseudo-code for the algorithm you used to answer part a) above.

Problem 6: Gordian

a) What are appropriate objective functions for a placement algorithm?
b) To what extent does Gordian accurately incorporate these objective functions? What are its deficiencies with respect to the objective functions?