GORDIAN ALGORITHM IMPLEMENTATION

- Languages Used: C++, MATLAB, python
- Qp solver: QuadProg (MATLAB)
  QuadProg (C++ library)
- GUI: python
- Partitioning Method
  - Recursive Bipartitioning
  - Slice (Extension)
- Global Optimizer
  - Original Gordian
  - GordianL (Extension to be finished)
- Area Minimization
Partition Method: Recursive Bipartitioning

- Odd levels: vertical cut
- Even levels: horizontal cut
- At each level $L$, there are $2^L$ partitions
- Each partition has at most 2 cells
- Maximum level = $\log_2(N)$
Method 1: Recursive Bipartitioning
industry 3 (438 IO cells, 14968 placeable cells)

Level 0  
Unconstrained  
~ 69 secs (CPU time ~27s)

Level 4  
16 partitions  
~ 4 mins (CPU time ~67s)

Level 13  
8192 partitions  
~ 13 mins (CPU time ~ 171s)
Method 1: Recursive Bipartitioning
p2 (102 IO cells, 1850 placeable cells) Level = 10

Level 10 Gordian Placement
Standard Cell Placement
Overlap Removal
Partition Method: Slice

- n cuts in total
- n/2 vertical cuts
- n/2 horizontal cuts
- Vertical cuts first from left to right
- Horizontal cut from bottom to top
Method: Slice
Fract (13 IO cells, 136 Cells)

Level 0  Level 4  Overlap removal  Wire connection
Placement result (fract)

recursive bipartitioning
WL$^2$: 211,812 um$^2$ ~ 6% longer
WL: 11,830 um ~ 3% longer

slice
WL$^2$: 199,922 um$^2$
WL: 11,491 um
Placement result (p1)

recursive bipartitioning
WL$^2$: 4,678,866 um$^2$
WL: 131,018 um

slice
WL$^2$: 7,284,720 um$^2$ ~ 56% longer
WL: 166,716um ~ 27% longer
## Best possible result

<table>
<thead>
<tr>
<th>Circuit</th>
<th># non-IO Cells</th>
<th>Run time</th>
<th>Wirelength² (um²)</th>
<th>Wirelength (um)</th>
<th>Area (um²)</th>
<th>Optimal Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>fract</td>
<td>136</td>
<td>~ 10 secs</td>
<td>199,922</td>
<td>11,491</td>
<td>882</td>
<td>6</td>
</tr>
<tr>
<td>p1</td>
<td>735</td>
<td>~ 10 secs</td>
<td>4,678,866</td>
<td>131,018</td>
<td>4,532</td>
<td>8</td>
</tr>
<tr>
<td>structP</td>
<td>1,850</td>
<td>~ 12 secs</td>
<td>16,892,461</td>
<td>290,515</td>
<td>11,431</td>
<td>5</td>
</tr>
<tr>
<td>p2</td>
<td>2,826</td>
<td>~ 30 secs</td>
<td>81,578,695</td>
<td>1,328,537</td>
<td>17,466</td>
<td>10</td>
</tr>
<tr>
<td>biomedP</td>
<td>6,228</td>
<td>~ 4 mins</td>
<td>7,766,064,773</td>
<td>85,929,200</td>
<td>38,420</td>
<td>10</td>
</tr>
<tr>
<td>industry2</td>
<td>12,237</td>
<td>~10 mins</td>
<td>6,292,839,028</td>
<td>63,653,951</td>
<td>75,738</td>
<td>12</td>
</tr>
<tr>
<td>industry3</td>
<td>14,968</td>
<td>~9 mins</td>
<td>921,535,358</td>
<td>10,433,707</td>
<td>92,967</td>
<td>10</td>
</tr>
</tbody>
</table>
To be finished

GordianL implementation