Project

One of:

• Pick a sizable parallel problem to implement
  ○ Ok to form a team of 2 students

• Pick a new language and implement past exercises

• Present a new language for parallelism in class
Programming Project Artifacts

• Description of the problem
  ○ Include an explanation of expected speedup (i.e., parallel versus inherently sequential)

• What you expected to learn and did learn from the project
  ○ Must not be so generic that it would apply to any HW
  ○ Must address learning about parallelism

• Implementation

• Measured speedup for implementation
  ○ Document measurements: platform, $P$, etc.
  ○ Speedup must be $> 1$
Due Dates

Due Friday, December 5:

• Description of the problem
• What you expect to learn from the project

Due Friday, December 12:

• All 4 parts of the completed project — even the parts already submitted
## Project Ideas from the Book

Implement existing parallel algorithms

e.g., Batcher’s Sort

Re-implement existing parallel benchmarks

Parallelize some useful computation

<table>
<thead>
<tr>
<th>Chess-End Games</th>
<th>3-Satisfiability Problems</th>
<th>Gene Sequence Alignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segmented Least-Squares</td>
<td>Video Motion Detection</td>
<td>A* Path-Finding for Games</td>
</tr>
<tr>
<td>Audio Analysis with a GPU</td>
<td>KD-Tree Construction</td>
<td>Image Convolution</td>
</tr>
<tr>
<td>Exact String Matching</td>
<td>MP3 Fast Fourier Transformation</td>
<td>Boid Simulation</td>
</tr>
<tr>
<td>Kohonen Maps</td>
<td>Ray Tracing</td>
<td>Galaxy Simulation</td>
</tr>
<tr>
<td>Prime Factorization</td>
<td>Rectangular Partitioning</td>
<td>Kenser-Ney Smoothing</td>
</tr>
<tr>
<td>Data Encryption</td>
<td>Checkers Min/Max Search</td>
<td>Artificial Neural Nets</td>
</tr>
<tr>
<td>Ray Casting</td>
<td>Julia Sets</td>
<td>Constraint Satisfaction</td>
</tr>
<tr>
<td>Sample Sort</td>
<td>Traveling Salesman Problem</td>
<td>Collaborative Filtering</td>
</tr>
</tbody>
</table>