**Good To Know:**
- Quiz Solutions on Canvas
- Tutoring Wed night

<table>
<thead>
<tr>
<th>Approach</th>
<th>Divide + Conquer</th>
<th>Dynamic Prog</th>
<th>Greedy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Divide into equal sized parts</td>
<td>Recursive thinking but store subproblems in array</td>
<td>Sort by an easy to determine order</td>
</tr>
<tr>
<td>Proof</td>
<td>Strong Induction (loop invariant)</td>
<td>Loop Invariant</td>
<td>Exchange</td>
</tr>
<tr>
<td>Runtime</td>
<td>Master Method</td>
<td>?</td>
<td>O(nlogn) usually</td>
</tr>
</tbody>
</table>

**Loop Invariants**

1. Create test invariant.
2. Check if terminations gives result. If not
3. Check if base case is true. If not
4. Check if maintenance is easy to prove. If not
Greedy Exchange

\[(\ldots j, k \ldots) \rightarrow (\ldots k, j \ldots)\]

Arbitrary exchange.
Not always possible/best exchange

Shipping Container

Greedy

![Graph](image)

Amount

items → decreasing value

Optimal

Change something about optimal to make more like greedy.
Good strategy: look at first place where differ differs
Won't ask you to find optimal greedy alg
- Give → argue optimal
- Create a greedy alg and show not optimal