S. KIMMEL

TALK THURS @ 12:30
MBH 220

Good To Know:

- Quiz Solutions on Canvas
- Tutoring Wed night

Not on exam
MWIS on exam

	Divide + Conquer	Dynamic Prog	Greedy
Approach	• Divide into equal sized parts • Recurse on parts • Combine	• Recursive thinking, but store subproblems in array	• Sort by an easy to determine order
Proof	• Strong Induction (loop invariant)	• Loop Invariant	• Exchange
Runtime	• Master Method	?	• $O(n \log n)$ usually

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

usually: add extra invariant

S.KIMMEL

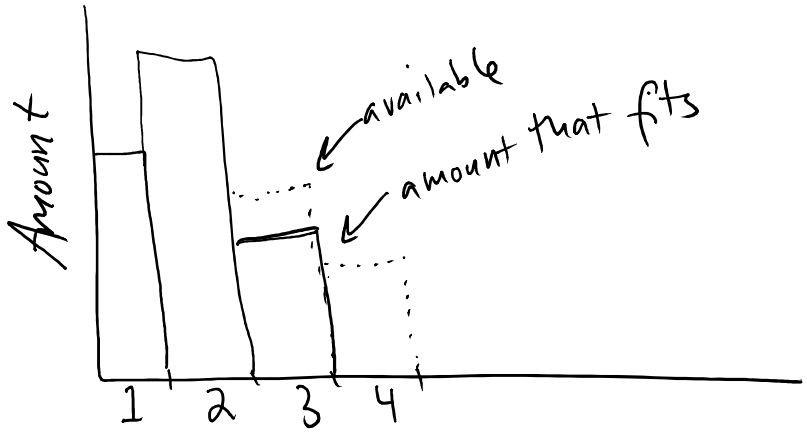
Greedy Exchange

$(\ldots j, k \ldots) \rightarrow (\ldots k, j \ldots)$

Arbitrary exchange.
Not always possible/best exchange

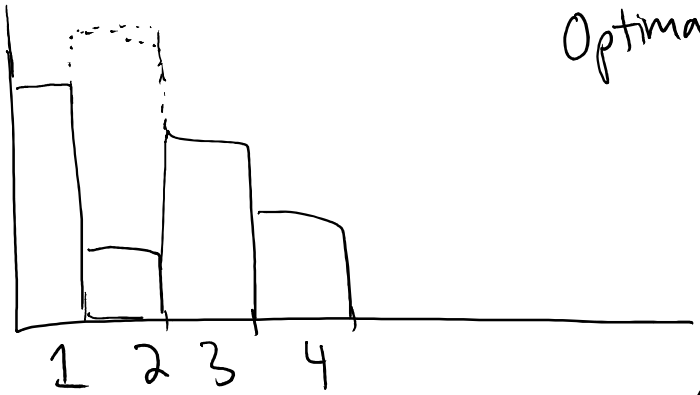
Shipping Container

Greedy



items → decreasing value

Optimal



Change <u>something</u> about optimal to make more like greedy.
Good strategy: look at first place where differ differs

S.KIMMEL

Won't ask you to find optimal greedy alg

- Give → argue optimal
- Create a greedy alg and show not optimal