1. The Cut Property for Minimum Spanning Tree
   (a) For any Minimum Spanning Tree and any partition of the nodes into two pieces, a minimum cost edge crossing from one piece to the other must be included in the tree.
   (b) Proof is by an “exchange argument”—for any tree that does not include such an edge, we can reduce the cost by swapping a minimum cost edge in for some other edge of the tree.

2. Given the Cut Property, the analysis of the correctness of both Prim’s Algorithm and Kruskal’s Algorithm are straightforward: the final tree will always have to include the chosen edge.
   (a) The proof is easier if we assume the costs are all distinct. This is easily ensured by “perturbing” the costs by a negligible amount so that they become distinct, even if they were originally not.

3. New problem: Knapsack
   (a) Given a set of items with an associated weight and value, and a maximum weight $W$, choose a subset with total weight at most $W$ of maximum value.
   (b) Natural greedy approaches fail, no obvious polynomial time algorithm ⇒ try dynamic programming
   (c) Key idea: introduce a new parameter to index the subproblems, the remaining capacity of the knapsack