

Chapter 1

Lecture 2

1.1 Exchange Argument

Consider the set of optimal solutions. We want to show that the greedy solution is in this set. To do this we can use proof by contradiction.

Proof. Suppose to the contrary, the greedy solution is not in this set of optimal solutions. Let OPT be an optimal solution that is **most similar** to Greedy. \square

Claim

We can make OPT more like the Greedy solution without making it worse.

If we can prove this claim then we have a contradiction because OPT is the **most** greedy optimal solution, and making it more like Greedy without making it worse means it is still an optimal solution closer to Greedy which is a contradiction.

The textbook's solution forks after the claim and creates another optimal solution that is more similar to Greedy, and repeats this until it reaches Greedy without making the solution worse. This is similar to an inductive approach.

1.2 Minimum Spanning Trees

Definition: MST

Given: graph $G = (V, E)$ with edge costs $c_e > 0$.

Goal: Cheapest set of edges such that all of V is connected.

Interestingly enough, MST are not a great example problem for Greedy Algorithms because so many algorithms turn out to work. We will prove three, but first we will prove two properties that will make proving the algorithms much easier.

Definition: Cut Property

If edge e is the minimum cost edge in a cut $(S, V - S)$, then e is in every MST.

We will refer to f from this point on to be the maximum cost edge in a cut.

Proof. Take some MST T^* and lightest edge e . Suppose to the contrary, edge e is not in T^* . What we will do is change T^* so that it does take e and becomes cheaper, then argue this is a contradiction. We take $f \in T^*$ which is in this cut, and replace it with e . Now we have

$$T^* + e - f$$

which is cheaper than T^* . Now we must prove that this new tree is cheaper than T^* and is a spanning tree. Trivially, this tree is cheaper because $e < f$. Consider $T^* + e$. This has one cycle. Let f be the edge in this cycle crossing the cut. Look at $T^* + e - f$. It is now cheaper than T^* and is a spanning tree because there is a path due to the cycle.

□

Definition: Cycle Property

C is a cycle, e is the most expensive edge in the cycle, then e does not belong to any MST.

Proof. Let T^* be a minimum spanning tree. Suppose to the contrary that $e \in T^*$. If you were to remove edge e , then T^* breaks into two disconnected trees. Consider $T^* - e + f$. This is cheaper than T^* . $T^* - e + f$ is a spanning tree because a cycle exists.

□

Now that we have proved these properties, we can use these in the rest of the course without proving them. Now we can prove three MST algorithms with them.

Definition: Kruskal's Algorithm

Put in the next closest cheapest edge. Order of edges by increasing cost, and add them to the tree as long as it does not produce a cycle. Runtime: $O(m \log n)$

Proof. By cut property, we can create a cut between one of the connected components and the set of vertices of the rest of the graph that does not create a cycle.

□

Definition: Prim's Algorithm

Start with root, and add cheapest edge from S where S are the set of nodes connected so far. Runtime: $O(m \log n)$

Proof. By cut property, we can create a cut between the set of all connected vertices and the set of all unconnected vertices and the added edge would be the minimum cut of this cut.

□

Definition: Reverse - Delete Algorithm

Start with all the edges and delete the most expensive edge without disconnecting the graph (assume that all c_e are distinct).

Proof. If there is an edge that is not part of a cycle, it disconnects the graph. If it is not, remove the most expensive edge by the cycle property. \square