Lecture 8: Greedy algorithm

In this class, we first finish the strongly connected component algorithm. This concludes the basic graph traversal.

The main topic of this lecture and a couple of following lectures are about greedy algorithm. Greedy algorithms make locally optimal decisions, which sometimes lead to globally optimal solution. Our first example is the coin change problem. You are given unlimited supplies of several coins (say quarters, dimes, nickels and pennies), you want to give out \( n \) cents using the fewest number of coins. The common greedy strategy of using largest coins is sometimes globally optimal, as in the US coins. However, if we slightly change the coin types, the situation will change. Suppose we have quarters, dimes and pennies but no nickels, the greedy strategy does not work. This can be seen by \( n = 30 \) example.

Our next example is the interval scheduling problem (also called activity selection problem) in the textbook. In class, several strategies were proposed:

1. First come, first serve.
2. Picking shortest requests.

We showed counter-examples that these two strategies are not globally optimal.

I claim that choosing the requests with earliest finish time is optimal. In class, I showed how to prove this strategy is optimal. This is done in two steps. This needs considering an optimal solution \( O \), where \( O = \{ o_1, \ldots, o_k, o_{k+1}, \ldots \} \). We denote the solution found by greedy algorithm is \( A = \{ a_1, \ldots, a_k \} \). Note that if \( O \) and \( A \) contain the same number of requests, we are done. Otherwise, we assume \( o_{k+1} \) exists. Note: comparing the greedy solution (\( A \) in this problem) and some optimal (unknown but must exist) solution (\( O \) in this case) is a frequently used scheme in proving the correctness of the greedy algorithm. Please carefully review the proof process. The outline is given below.

First, I show that \( f(a_i) \leq f(o_i) \) for each \( 1 \leq i \leq k \). This is proved by induction. When \( i = 1 \), it is true because the greedy algorithm picks the one with smallest finish time. Then we assume the claim holds for all indices less than \( i \). We now show \( f(a_i) \leq f(o_i) \). The key to this is that if this is not true, the greedy algorithm would have selected \( o_i \) instead of \( a_i \) since \( o_i \) is compatible with the previous \( a_{i-1} \) (why? Make sure you understand this).

Then, I show that \( o_{k+1} \) can not exist. Otherwise, \( f(a_k) \leq f(o_k) \), which will suggest the greedy algorithm will not stop picking after \( a_k \).

Then I also covered the problem of scheduling all intervals (jobs). See pages 122 to 125 for more details.