CSE 441T/541T: Advanced Algorithms **Fall Semester, 2004**

Overview of Greedy Choice/Optimal Substructure Methodology

September 7, 2004

1. Prove that the problem has the **greedy choice property** which says that the first step (call it g) taken by the greedy algorithm part of *some* optimal solution. This is typically done using a proof by contradiction in which some arbitrary optimal solution S that doesn't include g is modified to create a solution S' that includes g. It is very important to prove that S' is a legal solution and that S' is at least as good as S.

2. Prove that the problem has the **optimal substructure property**. Namely, let P be the original problem to be solved, let g be the first step taken by the greedy algorithm, and let S be an optimal solution for P that includes g (which by the greedy choice property must exist). The problem has the optimal substructure property if

   (a) There exists a subproblem P' of P that remains after g is included in the greedy solution. Be careful in this step that you truly have a subproblem. Think of defining your problem via the parameters of a recursive procedure. Now define P' by a recursive call. If you want to add any extra constraints then you need to generalize the form of the problem (i.e. add additional parameters). If you do this, then you have to go back and prove that the greedy choice property holds for this general problem form.

   (b) P' is optimally solved in S. That is, the solution for P' contained within S is optimal for P'. To do this it is best to express the value of S as some function that depends on the value of the optimal solution S' for P' and parameters of the problem. Then argue that P' is optimally solved within S.

Once you have proven that the greedy choice and optimal substructure properties hold then you are done. Below is a inductive proof of correctness that is not specific to the given problem. So you can always use this and thus there is no need to include it in your solutions. *But it is important that you understand how this proof works.*

The proof uses induction on the number k of steps in an optimal solution. Let $P(k)$ be the proposition that the greedy algorithm outputs on optimal solution when k steps are taken.

Basis Step: $P(1)$ holds. By the greedy choice property the first step, g, of the greedy algorithm is part of an optimal solution and since the optimal solution consists of only 1 step it follows that g by itself is an optimal solution.

Inductive Step: We must show that $\forall k \geq 1\ P(k) \rightarrow P(k+1)$. Consider an input P for which the optimal solution consists of $k+1$ steps. By the greedy choice property, g is part of some optimal solution S. By the optimal substructure property (2a), it follows that after g is placed into the greedy solution there remains a subproblem P'. By the optimal substructure property (2b) it also follows that the optimal solution S for P can be obtained by combining g with an optimal solution S' for P'. So if S consists of $k+1$ steps then S' consists of k steps. Thus by the inductive hypothesis, the greedy algorithm solves P' correctly which completes the inductive step.

Note that just like there can be more than one greedy algorithm for a problem, for a particular greedy algorithm there can be more than one way to decompose the choices of the greedy algorithm into steps. Some ways can lead to easier proofs than others. Come see us if you have any questions about this. In your class notes, the textbook, and the practice problems available on the course homepage (under handouts) you can find lots of examples of proving that the greedy choice and optimal substructure properties hold. You will notice that these proofs require you to think about the particular problem at hand and can't just be done by "turning a crank" without thinking about what you are doing.