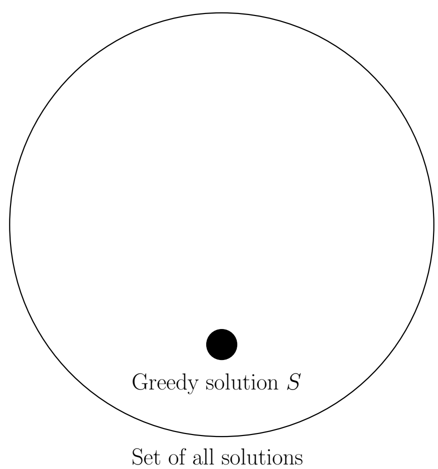
Strategies for Proving Greedy Algorithms are Correct

CMSC 27230: Honors Theory of Algorithms

1 Challenge of proving a greedy algorithm is optimal

The greedy algorithm gives us one solution S_{greedy}. However, the set of all possible solutions is extremely large. How do we know that there is no better solution S' somewhere within this set?



1.1 The greedy algorithm stays ahead

One way to show that the greedy solution is optimal is to show that at each step, it does at least as well as any other solution. The kind of statement we need to show is as follows (where the function f depends on the problem):

Lemma 1.1. *For all solutions S' and all $j \geq 1$, $f_j(S_{greedy}) \geq f_j(S')$*

Proof strategies. Let S' be another solution. We can prove that $\forall j \geq 1(f_j(S_{greedy}) \geq f_j(S'))$ both by induction and by a proof by contradiction.

To prove that $\forall j \geq 1(f_j(S_{greedy}) \geq f_j(S'))$ by induction, we need to do the following:

Base case: Show that $f_1(S_{greedy}) \geq f_1(S')$

Inductive step: Show using the nature of the greedy algorithm that for all $j \geq 2$, if $f_{j-1}(S_{greedy}) \geq f_{j-1}(S')$ then $f_j(S_{greedy}) \geq f_j(S')$

To prove that $\forall j \geq 1(f_j(S_{greedy}) \geq f_j(S'))$ using a proof by contradiction, we can let j be the first index such that $f_j(S_{greedy}) < f_j(S'))$. We then have the following two cases:

1. $j = 1$. To obtain a contradiction here, we must show directly that $f_1(S_{greedy}) \geq f_1(S')$.
2. $j > 1$. In this case, since we took j to be the first index such that $f_j(S_{greedy}) < f_j(S')$ we know that $f_{j-1}(S_{greedy}) < f_{j-1}(S')$. We can then use this and the nature of the greedy algorithm to show that $f_j(S_{greedy}) \geq f_j(S')$, which is a contradiction.

□

Remark 1.2. *While these two arguments are phrased differently, they are essentially the same. In particular, for both arguments we need to show that $f_1(S_{greedy}) \geq f_1(S')$ and we need to show that if $j \geq 2$ and $f_{j-1}(S_{greedy}) \geq f_{j-1}(S')$ then $f_j(S_{greedy}) \geq f_j(S')$.*

While this gives the general structure of a proof that the a greedy algorithm stays ahead, you should tailor the argument to the specific problem in question. In particular, you should be sure to specify exactly what the measure f_j of progress is.

1.2 Exchange arguments

A second way to show that S_{greedy} is indeed an optimal solution is to show that for any optimal solution S', we can iteratively adjust S' to make it closer and closer to S_{greedy} until it becomes S_{greedy} and moreover, each of these adjustments either gives us an equally good solution or makes our solution even better.

The intuition for why this implies that S_{greedy} is optimal is as follows. Since each of the adjustments either gives us an equally good solution or makes our solution even better, the solution we end with must be at least as good as the solution we started with. Since we started with S' and ended with S_{greedy}, this shows that S_{greedy} is at least as good as S'. Since S' is optimal, we must have that S_{greedy} has the same value as S' and is also optimal.

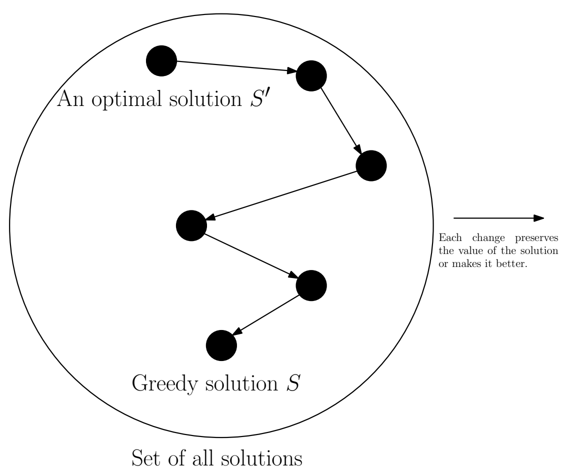


Figure 1: This figure illustrates the argument for showing that S_{greedy} is optimal.

Remark 1.3. *The adjustments in this kind of argument generally involve exchanging elements of S' to make it closer to S_{greedy}, which is why this kind of argument is generally called an exchange argument.*

We now show how to make this kind of argument more precise. For simplicity, we relabel the elements so that the greedy solution is $S_{greedy} = (1, 2, \ldots, n)$.

Theorem 1.4. *If we can show that whenever $S = (1, 2, \ldots, k-1, s_k, s_{k+1}, \ldots, s_n)$ is an optimal solution, the solution S' obtained by swapping s_k and k is also an optimal solution then $S_{greedy} = (1, 2, \ldots, n)$ is an optimal solution.*

Proof. One way to prove this is by using induction.

Lemma 1.5. *For all $k \in [0, n]$, there is an optimal solution S such that the first k elements of S are $1, 2, \ldots, k$.*

Proof. The base case $k = 0$ is trivial as we can take S to be any optimal solution. For the inductive step, assume that there is an optimal solution $S = (1, 2, \ldots, k, s_{k+1}, s_{k+2}, \ldots, s_n)$ such that the first k elements of S are $1, 2, \ldots, k$. Letting S' be the solution obtained by swapping s_{k+1} and $k+1$, S' is an optimal solution such that the first $k+1$ elements of S' are $1, 2, \ldots, k, k+1$, as needed. □

Applying this lemma with $k = n$ proves that $S_{greedy} = (1, 2, \ldots, n)$ is an optimal solution.

Another way to prove this is using a proof by contradiction. Assume that $S_{greedy} = (1, 2, \ldots, n)$ is not optimal. Let $k \in [n]$ be the minimum number such that there is no optimal solution which starts with $1, 2, \ldots, k$.

By the minimality of k, there exists an optimal solution $S = (1, 2, \ldots, k-1, s_k, s_{k+1}, \ldots, s_n)$ which starts with $1, 2, \ldots, k-1$. By the assumption in the theorem statement, letting S' be the solution obtained by swapping s_k and k, S' is also an optimal solution. Moreover, S' starts with $1, 2, \ldots, k$. However, this is impossible as we choose k to be the minimum number such that there is no optimal solution which starts with $1, 2, \ldots, k$. □

Theorem 1.6. *If we can show that whenever $S = (s_1, \ldots, s_n)$ is an optimal solution, $j \in [n-1]$, and $s_j > s_{j+1}$, the solution $S' = (s_1, \ldots, s_{j-1}, s_{j+1}, s_j, s_{j+2}, \ldots, s_n)$ obtained by swapping s_j and s_{j+1} is also an optimal solution then $S_{greedy} = (1, 2, \ldots, n)$ is an optimal solution.*

Proof strategies. One way to prove this statement is with a proof by contradiction. Assume S_{greedy} is not an optimal solution and let $S = (s_1, \ldots, s_n)$ be the optimal solution which is first in the lexicographic order. Since $S \neq S_{greedy}$, there is an index $j \in [n-1]$ such that $s_j > s_{j+1}$. By the assumption in the theorem, the solution $S' = (s_1, \ldots, s_{j-1}, s_{j+1}, s_j, s_{j+2}, \ldots, s_n)$ obtained by swapping s_j and s_{j+1} is also an optimal solution. However, S' comes before S in the lexicographic ordering which contradicts how we chose S. Thus, S_{greedy} must be an optimal solution.

Alteratively, we can prove this statement by induction on the number of inversions (where an inversion is a pair i and j such that $i < j$ and $s_i > s_j$).

Lemma 1.7. *For all $k \in [0, \binom{n}{2}]$, there is an optimal solution with at most k inversions.*

Proof. Here the base case $k = \binom{k}{2}$ is trivial. Assume that there is an optimal solution $S = (s_1, \ldots, s_n)$ with at most k inversions. If S has no inversions then we are done. Otherwise, there must exist a $j \in [n-1]$ such that $s_j > s_{j+1}$. By the assumption in the theorem, the solution $S' = (s_1, \ldots, s_{j-1}, s_{j+1}, s_j, s_{j+2}, \ldots, s_n)$ obtained by swapping s_j and s_{j+1} is also an optimal solution. Moreover, it is not hard to show that S' has exactly one fewer inversion than S. This means that S' has at most $k-1$ inversions, as needed. $\square$

The theorem follows by applying Lemma 1.7 with $k = 0$. $\square$