1 Basic Definitions

A *Problem* is a relation from input to acceptable output. For example,

INPUT: A list of integers $x_1, \ldots, x_n$

OUTPUT: One of the three smallest numbers in the list

An algorithm A solves a problem if A produces an acceptable output for EVERY input.

A *optimization problem* has the following form: output a best solution S satisfying some property P. A best solution is called an *optimal solution*. Note that for many problems there may be many different optimal solutions. A *feasible solution* is a solution that satisfies the property P. Most of the problems that we consider can be viewed as optimization problems.

2 Proof By Contradiction

A proof is a sequence $S_1, \ldots, S_n$ of statements where every statement is either an axiom, which is something that we've assumed to be true, or follows logically from the precedding statements.

To prove a statement p by contradiction we start with the first statement of the proof as $\bar{p}$, that is not p. A proof by contradiction then has the following form

$$\bar{p}, \ldots, q, \ldots, \bar{q}$$

Hence, by establishing that $\bar{p}$ logically implies both a statement q and its negation $\bar{q}$, the only way to avoid logical inconsistency in your system is if p is true.

Almost all proofs of correctness use proof by contradiction in one way or another.

3 Exchange Argument

Here we explain what an exchange argument is. Exchange arguments are the most common and simpliest way to prove that a greedy algorithm is optimal for some optimization problem. However, there are cases where an exchange argument will not work.

Let A be the greedy algorithm that we are trying to prove correct, and $A(I)$ the output of A on some input I. Let O be an optimal solution on input I that is not equal to $A(I)$.

The goal in exchange argument is to show how to modify O to create a new solution O' with the following properties:

1. O' is at least as good of solution as O (or equivalently O' is also optimal), and

2. O' is "more like" $A(I)$ than O.

Note that the creative part, that is different for each algorithm/problem, is determininig how to modify O to create O'. One good heuristic to think of A constructing $A(I)$ over time, and then to look to made the modification at the first point where A makes a choice that is different than what is in O. In most of the problem that we examine, this modification involves changing just a few elements of O. Also, what "more like" means can change from problem to problem. Once again, while this frequently works, there's no guarantee.

4 Why an Exchange Argument is Sufficient

We give two possible proof techniques that use an exchange argument. The first uses proof by contradiction, and the second is a more constructive argument.

Theorem: The algorithm A solves the problem.

Proof: Assume to reach a contradiction that A is not correct. Hence, there must be some input I on which A does not produce an optimal solution. Let the output produced by A be $A(I)$. Let O be the optimal solution that is most like $A(I)$.

If we can show how to modify O to create a new solution O' with the following properties:

1. O' is at least as good of solution as O (and hence O' is also optimal), and

2. O' is more like $A(I)$ than O.

Then we have a contradiction to the choice of O.

End of Proof.

Theorem: The algorithm A solves the problem.

Proof: Let I be an arbitrary instance. Let O be arbitrary optimal solution for I. Assume that we can show how to modify O to create a new solution O' with the following properties:

1. O' is at least as good of solution as O (and hence O' is also optimal), and

2. O' is more like $A(I)$ than O.

Then consider the sequence $O, O'', O''', O'''', \ldots$

Each element of this sequence is optimal, and more like $A(I)$ than the proceding element. Hence, ultimately this sequence must terminate with $A(I)$. Hence, $A(I)$ is optimal.

End of Proof.

I personally prefer the proof by contradiction form, but it is solely a matter of personal preference.

5 Proving an Algorithm Incorrect

To show that an algorithm A does not solve a problem it is sufficient to exhibit one input on which A does not produce an acceptable output.

6 Maximum Cardinality Disjoint Interval Problem

INPUT: A collection of intervals $C = \{(a_1, b_1), \ldots, (a_n, b_n)\}$ over the real line.

OUTPUT: A maximum cardinality collection of disjoint intervals.

This problem can be interpretted as an optimization problem in the following way. A feasible solution is a collection of disjoint intervals. The measure of goodness of a feasible solution is the number of intervals.

Consider the following algorithm A for computing a solution S:

1. Pick the interval I from C with the smallest right endpoint. Add I to S.

2. Remove I, and any intervals that overlap with I, from C.

3. If C is not yet empty, go to step 1.

Theorem: Algorithm A correctly solves this problem.

Proof: Assume to reach a contradiction that A is not correct. Hence, there must be some input I on which A does not produce an optimal solution. Let the output produced by A be $A(I)$. Let O be the optimal solution that has the most number of intervals in common with $A(I)$.

First note that $A(I)$ is feasible (i.e. the intervals in $A(I)$ are disjoint).

Let X be the leftmost interval in $A(I)$ that is not in O. Note that such an interval must exist otherwise $A(I) = O$ (contradicting the nonoptimality of $A(I)$), or $A(I)$ is a strict subset of O (which is a contradiction since A would have selected the last interval in O).

Let Y be the leftmost interval in O that is not in $A(I)$. Such an interval must exist or O would be a subset of $A(I)$, contradiction the optimality of O.

The key point is that the right endpoint of X is to the left of the right endpoint of Y. Otherwise, A would have selected Y instead of X.

Now consider the set $O' = O - Y + X$.

We claim that:

1. O' is feasible (To see this note that X doesn't overlap with any intervals to its left in O' because these intervals are also in $A(I)$ and $A(I)$ is feasible. And

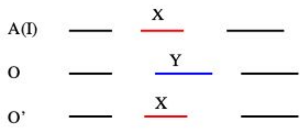

Figure 1: The instances $A(I)$, O and O'

X doesn't overlap with any intervals to its right in O' because of the key point above and the fact that O was feasible.),

2. O' has as many intervals as O (and is hence also optimal), and

3. O' has more intervals in common with $A(I)$ than O.

Hence, we reach a contradiction.

End of Proof.