David Ogle

11 March 2013

ECS FYS

Greedy Algorithms

Greedy algorithms generally involve finding the cheapest or simplest way to go about solving a problem at hand. The most common example used to illustrate a greedy algorithm would be through use of a railroad or other branch-like structures that must connect. Through use of the greedy algorithm, the railway, or branches, would connect by going between each city, or point, taking the cheapest route and the route that would not create a cycle. Cycles are when the cheapest route would cause the railway to connect with a city already connected in some other way, thus if this were to happen the second cheapest route would be taken. The algorithm would continue to find the cheapest routes until every point is connected in some way to the others without creating any unnecessary cycles. This is just one type of greedy algorithm which is commonly referred to as the minimal spanning tree. There is a method that was created for greedy algorithms using the minimal spanning tree named Kruskal's method. Kruskal's minimal spanning tree algorithm was basically ordering the connections between points in ascending order from left to right and then going through and ridding of any that would cause a cycle. This method is meant to find the absolute minimum value one could achieve through picking cheapest values between points. Another example of this would be to look at the BART in San Francisco. BART stands for Bay Area Rapid Transit and there are a few different tracks, but for the sake of this argument just one will be looked at. Following one train, one can see that BART goes simply through every place in a straight line and does not do any cycles. This is a sort of minimal spanning tree in that the simplest way was found to connect all the destinations which may or

may not have been cheaper but it is likely that it was when following the idea of greedy algorithms.

Greedy algorithms are a great way to find a simple solution that is cheap and effective for the time being. However, there is a large setback with using these algorithms since they will not always produce the best results in the long run or even possibly the near future. In the case of BART and the railways, it is likely an acceptable approach since it is widely used with almost all train tracks and transports. In a separate case, like a game of chess for example, playing by choosing the simplest move each turn will end up being a big problem in the long run, especially when paired against someone who is planning ahead and making small moves all building up to a big move to win the game. Greedy algorithms were not meant to beat a game of chess or even something like tic tac toe, however there are many turn-based games where these algorithms would work out either all the time or most of those times. A turn-based game revolves around choosing the best option at the time being rather than worrying about five turns ahead. Therefore, they fit the description of a greedy algorithm rather well with only a slight change being that these turn-based games may not always work by choosing the simplest method at the time being. In a game of Monopoly, for example, one has very simple decisions to make such as buy the property, do not buy the property, pay $50 to get out of jail, or take the chance to roll doubles. Following the greedy algorithm, it would be simpler and cheaper to pay the $50 to get out now since you will get back into the action and also not have to take any chances and waste time. However, the first two options might not always work by following the greedy algorithm since the simplest method would vary depending on the situation at hand and that option might end up being bad in the long run. For example, if the person would lose almost all their money buying the property, it would be wise not to buy it, but if the next person only needed that property in

order to get a monopoly then in the long run the first person might regret not buying it when they had the chance. In the end, it can be seen that greedy algorithms must not be used lightly and can have disastrous effects in the long run even if they seem to be appropriate in the short run. Just because they are to be used with caution does not mean that they cannot be used, since at times the greedy algorithm can actually be the best way in short and long run since it can save money and be a simple and quick solution to problems.