Greedy for Contests

Daniel D. Kang

November 18, 2010

1 Introduction

Like dynamic programming, greedy is essentially an optimization algorithm. Greedy algorithms work by choosing a *local* optimal choice (the greedy choice) to find the *global* optimum. They are efficient, powerful, and easy to code. However, they can only be used in certain situations. The hardest part of applying a greedy algorithm is simply realizing that a problem is greedy. The most common error is applying greedy where it does not apply. I repeat: the most common error is applying greedy where it does not apply. Many problems appear to be greedy, but simply aren't. If you can prove a greedy solution, then use it, but be careful that the solution is actually greedy.

2 Worked Example

The classic example of greedy the fractional knapsack problem:

Problem: A robber enters a store with a knapsack that can hold n pounds of merchandise. There are k boxes of donuts in the store. For each box i, he can take a donut individually without taking the entire box. Each box holds donuts of a different weight, w_i, and different value, v_i. What is the greatest value of merchandise he can steal and hold in his knapsack? Note: The robber is not confined by the bounds of human decency, so it IS possible for him to split up a donut (that monster).

Solution: The solution would be simply to take the donuts from the highest value-to-pound ratio first. This ensures we have most value per pound, thus ensuring an optimal solution.

Notice that this solution takes local maxima (the highest value-to-pound ratio at each step) to find the optimal solution.

3 Example

This is another classical example of the greedy algorithm:

Problem: We have n tasks that need to use a resource (for example, a lecture hall) which can only be used by only one activity at a time. Each activity i starts at time s_i and ends at time f_i, where $s_i \leq f_i$. What is the maximum number of activities you can run?

Solution: Sort the activities by increasing finish time, breaking ties arbitrarily. Iterate over the sorted list, beginning with the first one, and choose the next compatible activity.

Convince yourself that the solution is optimal.

4 Issues With Greedy

As stated in the introduction, *solutions often appear to be greedy, but aren't.* Consider the following example:

Problem: Given the denominations of coins for a newly founded country, the Dairy Republic, and some monetary amount, find the smallest set of coins that sums to that amount. The Dairy Republic is guaranteed to have a 1 cent coin.

Greedy Solution: Take the largest coin value that isn't more than the goal and iterate on the total minus this value.

This solution is clearly wrong. Find a counterexample. The correct solution is a dynamic programming solution.

5 Practice Problems

1. Farmer John has leashed his N ($1 \leq N \leq 50,000$) cows to stakes (at integer locations) next to a linear fence. Every cow is straining her leash as much as she can to the east (though never beyond the end of the fence). FJ's wife can locate herself halfway between two integer points along the fence and cut all the leashes that cross in front of her at that place. Given the length of each leash and the location of its stake in the ground, what is the minimal number of cuts she must perform to free the cows? (Dean, Kolstad, 2006)

2. To avoid burns while tanning, each of the C ($1 \leq C \leq 50,000$) cows must cover her hide with sunscreen when they're at the beach. Cow i has a minimum and maximum SPF rating that will work. If the SPF rating is too low, the cow suffers sunburn; if the SPF rating is too high, the cow doesn't tan. The cows have L ($1 \leq L \leq 50,000$) bottles of sunscreen lotion, each bottle i with an SPF rating. Bottle i can cover $cover_i$ ($1 \leq cover_i \leq C$) cows with lotion. A cow may lotion from only one bottle. What is the maximum number of cows that can protect themselves while tanning given the available lotions? (Cox, 2001)

3. N ($1 \leq N \leq 50,000$) cows are driving in separate cars along a highway. Cow i can drive in any of M different highway lanes ($1 \leq M \leq N$) and can travel at a maximum speed of S_i ($1 \leq S_i \leq 1,000,000$) km/hour. The cows take extraordinary measures to avoid collisions. Cow i reduces its speed by D ($0 \leq D \leq 5,000$) km/hour for each cow in front of it (though never below 0 km/hour). A cow might actually travel faster than a cow directly in front of it. There is a minimum speed law which requires everyone on the highway to travel at a minimum speed of L km/hour. What is the maximum number of cows that can drive on the highway while obeying the minimum speed limit law? (Wu 2007)

4. FJ wants to repair a length of the fence around the pasture. He measures the fence and finds that he needs N ($1 \leq N \leq 50,000$) planks of wood, each having some integer length L_i units. He then purchases a single long board just long enough to saw into the N planks (i.e., whose length is the sum of the lengths L_i). FJ is ignoring the extra length lost to sawdust when a saw cut is made; you should ignore it, too. FJ doesn't own a saw, so he asks Farmer Don if he may borrow a saw. Farmer Don doesn't lend FJ a saw but instead offers to charge Farmer John for each of the $N-1$ cuts in the plank. The charge to cut a piece of wood is exactly equal to its length. (Cutting a plank of length 21 costs 21 cents.) Farmer Don then lets Farmer John decide the order and locations to cut the plank. What is the minimum amount of money he can spend to create the N planks? (Ho, 2006)

5. There is a long list of stalls, some of which need to be covered with boards. You can use up to N ($1 \leq N \leq 50$) boards, each of which may cover any number of consecutive stalls. Cover all the necessary stalls, while covering as few total stalls as possible. (1999 US Open)

6. N cows go through a milking queue. The first part of the process takes A_i seconds for cow i, cows go through a milking queue. The first part of the process takes B_i for cow i. Only one farmer is available for each part of the process, so it is feasible that the second farmer will be doing nothing while the first farmer is stuck milking a cow, or there can be a backup if the second farmer takes too long milking a cow (though the first farmer can continue milking). Determine in $O(NlogN)$ time the minimum time it takes to milk all the cows. (2006 US Open)