Math 320 Homework 1.9 part 2

Chris Rytting

September 22, 2015

Algorithm description

Consider the following problem. An individual, Charlie, is on a treasure-hunting expedition. He stumbles upon a vast treasure trove, where there is a large, but finite, amount of jewels and valuables. Charlie is faced with a problem, though. He can only carry a certain weight W before his knapsack breaks. Every item has its weight w_i and value v_i etched on the side. How does Charlie figure out what to take? He employs the following algorithm, implemented in python on his MacBook Pro.

He fills his knapsack with n random objects x_i such that $\sum_{i=1}^{n} x_i < W$. He then checks the ratio $\frac{v_i}{w_i}$ to see if this ratio is higher than the objects already in the backpack. This algorithm will have temporal complexity of $O(2n)$, since he will have to store every items ratio and replace the items in his pack a potentially huge number of times.

He then realizes that there is a better way! He decides not to store every ratio in memory. Rather, he only stores a ratio if it is a new, unseen ratio. This way, he stores a fraction of what he once stored, and only places the most valuable items in his knapsack. This algorithm has a temporal complexity of $O(n)$, since he only has to store a few values and only places the most valuable items in his knapsack, and Charlie goes home rich!

Fin