1 Background

In this project we are going to be taking a stroll down memory lane and create a CD of some of our favorite songs. However, we are very particular about the songs that we want on our CD, and we only want the very best of the best. We have a modest collection of 192 songs to choose from, but our two sided CD can only have 11 minutes and 32 seconds (this is 692 seconds) on side one, and 8 minutes and 27 seconds on side two (this is 507 seconds). Each song has been rated with a score $p_i \in [0, 100]$ and has length w_i seconds. Your job then is to find the best song selection that can fit in the CD so that the sum of all the scores of the songs on both sides of the CD is as large as possible.

2 Dynamic Programming

You will be given a list of songs s_i with their score $p_i \in [0, 100]$, and length w_i, for $i = 1, \ldots, 192$. Each song can only be on the CD at most once. Like we said previously, the song selection that you choose must be "optimal" meaning that your solution must maximize the sum of the ratings of all the songs on the CD. A good way to solve a problem like this is with a dynamic programming technique.
Let $Q_{i,j,k}$ be the optimal value that you can get when you try to fit the first k songs (out of the 192 that you will be given) into a CD whose first side can hold i seconds and the second side can hold j seconds. E.g., $Q_{20,30,15}$ will be the optimal value when you try to fit the first 15 songs into a CD whose first side can hold 20 seconds and whose second side can hold 30 seconds. We want you to compute the value $Q_{692,507,192}$, as well as which songs go on which side, achieving the optimal value $Q_{692,507,192}$. To derive $Q_{i,j,k}$, note that

1. You either include song k in side A (In this case $Q_{i,j,k} = Q_{i-w_k,j,k-1} + p_k$).
2. Or you include song k in side B (In this case $Q_{i,j,k} = Q_{i,j-w_k,k-1} + p_k$).
3. Or you do not include song k in the CD at all (In this case $Q_{i,j,k} = Q_{i,j,k-1}$).

3 Implementation

Now implement a dynamic programming algorithm that solves the above recursion. It will take in a file that is a list in the form "|artist|album|track|rating|length" where the "length" field is specified in seconds. This input file will be given to you. Your solution should give the list of titles used for each side, the total time used, and the total "score" of the CD.

Task 1: Implement the described program, where the available time for side one is 692 seconds and the available time for side two is 507 seconds.

4 Academic Integrity

This project is to be done individually. Any implementation or analysis ideas that are not purely your own should be cited, and in particular ideas taken from books or the Internet. Please do not collaborate with other students while developing your programs, and please abide by the student honor code.