EFE FURKAN EKMEKCİ IE343 TERM PROJECT.

In this project ı have created and coded 2 algorithms to solve our problem. First algorithm is for maximizing the objective. It is like a Knapsack algorithm goal is without exceeding the album duration, to put max value songs in our album.In line 139-147ı create a function named createTracks. I convert sequential data matrix to a double array. Then ı get all datas to in tracklist with coding tracks.add in line 140. Weight value sequential data will be in this track list. Then ı create a new arraylist called album and a function called algortihm 1 in lines 30-31. Algorithm1 will be on line 64-84. I set the capacity 30\*60. Then ı want my while to turn until capacity bigger than 0. I looked for maxvalues. For a better algorithm ı used value/second. I write a for loop to look all tracks when ı turning my while. If value/sec bigger than maxvalue and duration of this song is lower than my capacity maxIndex=i, maxvalue= value/sec. We will find in all songs we will find the best ratio song in unused tracks. I write a break for finish while. Else addtrack for album and update my capacity. In algorithm2 our goal is to find best order for our songs with max value point.We set a function. In lines 96-136 ı tried to make max value of a song when we listen after another song. I create a array called trackorder. To find max value track ı write largestA and largestB. First if works for trackvalue is bigger than LargestA which is our top valued song, change for new one and make largestA—> LargestB. B is my second best. Then ı get ids of that 2 tracks in line 110. Then update the ids after that ı get out this 2 songs from my id list. İn line 116 ı create a for loop. First for looks new arraylist without top2 tracks, second for looks to our album. Then there is a if here. Looks tracks of album and find a track in best value. İn line 122 ı make for loop and look track\_ids and try to find largest again and again. From 148- last ı taked from given codes.

These are my results. I have find the obj=529.3. I think it is good algorithm. We can see the track orders. Our runtime for first algorithm is 0.007sec. For all alogrithms 0.04sec.

![A picture containing text

Description automatically generated]()