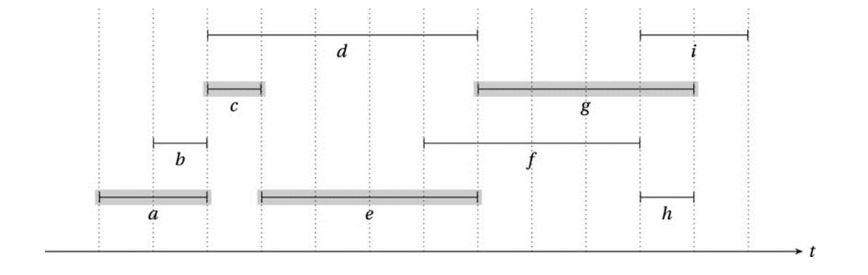
Timothy Indrieri

Brandon Artner

Edward Morkunas

Assignment 5

An example of an operating system problem (where there are n number of processes and the system sees where the next process finishes to have the next process placed) would be the activity selection process. The activity selection process has a start time, end time, and the profit of the schedule denoted as the following (S1,,F1, P1) respectively. Where the output is a set of activities that are most profitable.



(Source: Gekko, Gordon. "Greed Is Good? Prove It!" Python Algorithms. N.p., Fall 2014. Web. 19 Nov. 2016. )

The following diagram above shows the activities a,...,h where the horizontal axis is the time in which the activitIes start and finish. Whereas the vertical axis is the profit of the certain processes.

Our implementation of the activity selection process makes use of the algorithm described in the textbook. We first take in the input and read it in as a list of activities organized by ( start time, finish time , profit of activity). The list is then sorted. We then called a method that was defined called “selection”. This function performs the selection of the activities and will return a list of the indices of the corresponding activities. First the function will initialize a list named A which holds the max profits for our list. Then calls another function called “largestFinishTimes”. This function will find the largest finish times that are no greater than the start time of activity *i.* Selection will call another function called “findDistincts” that will sort the finish times of each activity and will ignore duplicates. Selection will then iterate through the list of distinct finish times denoted in the book as U1,...Un . An inner for loop will loop through the list of activities. It will then compare the current distinct finish time to every finish time inside the list of activities. If they match then we perform a check to see if the profits are a maximum. If we find a new max, we append the index into a list named J. After the inner for loop, we check to if the previous element at j-1 is larger than the max. If the previous element is larger than we are sure that the previous index holds the max profit. We then set the maximum profit accordingly and move to the next iteration.The following image holds the results of our test run with the list of activities.



The algorithm displays the list for debugging purposes. The output of our program is [ 1, 3 ]. These are the indices of the sorted list that will provide the maximum profit.