Chapter 6- Greedy Algorithms

2022

Prepared by: Beimnet G.

Greedy Algorithms

Algorithms for optimization problems typically go through a sequence of steps, with a set of choices at each step.

A greedy algorithm always makes the choice that looks best at the moment.

That is, it makes a locally optimal choice.

Activity Selection Problem

Problem Statement:

You are given n activities with their start and finish times. Select the maximum number of activities that can be performed by a single person, assuming that a person can only work on a single activity at a time.

Activity Selection Problem

	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
s_i	0	2	1	3	4	6	5	∞
f_i	0	3	4	5	6	8	9	∞

$S_{i,j}$ - the set if activities which starts after a_i finishes and finishes before a_j starts.

$A_{i,j}$ - the set of compatible activities in $S_{i,j}$ (the solution set)

Activity Selection Problem

If we know an activity a_k in in the optimal solution, we know that the final solution will contain all compatible activities in $S_{i,k}$ and $S_{k,j}$ and a_k.

What is a_k?

A[i,j]- the max number of activities in the optimal solution.

$$A[i,j] = \begin{cases} \max\limits_{i<k<j} \quad 1 + A[i,k] + A[k,j], & S_{i,j} \neq \emptyset \\ 0, & S_{i,j} = \emptyset \end{cases}$$

Activity Selection Problem: Greedy Approach

Making the Greedy Choice:

What if we could choose an activity to add to our solution without having to solve all the subproblems?

Intuition: If we pick an activity that finishes early it will free up the resource for more activities that follow it.

Activity Selection Problem: Greedy Approach

The greedy choice is to always pick the next activity whose finish time is least among the remaining activities and the start time is more than or equal to the finish time of the previously selected activity. We can sort the activities according to their finishing time so that we always consider the next activity as minimum finishing time activity.

1) Sort the activities according to their finishing time

2) Select the first activity from the sorted array and print it.

3) Do the following for the remaining activities in the sorted array.

Greedy Approach: Problems

1. Fractional knapsack problem
2. Egyptian fractions
3. Minimum spanning tree