Algorithms – handout 5

Greedy Algorithms

December 9, 1999

When searching for the optimal solution to a problem that has many feasible solutions, search may progress in one of several ways, depending on the structure of the problem. As a rough and incomplete classification, these are three of the major approaches:

1. *Exhaustive search:* Searches through all feasible solutions, or through a large fraction of them (where "large fraction" is a term that may depend on the context). Used for problems that have very little structure, or for which the structure is poorly understood. For example, for finding maximum cliques in graphs.

2. *Dynamic programming:* Searches through a very small fraction of the feasible solution. Typically, dynamic programming is an iterative algorithm. At each irteration, a small (polynomial size) set of partial solutions is maintained, one of which is known to lead to the optimal solution. In the next iteration, members of the set are extended based on information from other members of the set. At the last step, the optimal solution is chosen from the set. To keep the sizes of the sets small, the problem needs to have sufficient structure so that all partial solutions can be grouped into a small number of equivalent classes. For example, subset sum with polynomial weights.

3. *Greedy:* This may be viewed as the ultimate form of dynamic programming, in which only one partial solution is maintained. The problem needs to have much more structure for this approach to work. For example, minimum spanning trees.

Remark: There are other commonly used search strategies, such as *branch and bound.* Also, sometimes the method of finding the optimal solution involves ingredients that do not have intuitive explanations as search procedures. For example, finding triangles in graphs using fast matrix multiplication.

We shall study several problems that can be solved to optimality in polynomial time using the greedy algorithm.

Homework:
(Exercise 17.1-2 from CLR.) Give an efficient greedy algorithm to legally color interval graphs with the minimum number of colors, when the interval representation of the graph is given. (That is, the vertices of the graph correspond to intervals on the real line, and two vertices are connected by an edge, and hence need to receive different colors, if their respective intervals overlap.)