CSE421: Design and Analysis of Algorithms	May 3, 2019
Lecturer: Shayan Oveis Gharan	Lecture 15 Approximation Algorithms

The approximation ratio of an algorithm ALG is the ratio of the cost of the solution produced by the algorithm over the cost of the optimum solution in the *worst case*. In the field of approximation algorithms, we want to design a polynomial time algorithm with the smallest possible approximation ratio. In this lecture we will design approximation algorithms for two fundamental NP-hard problems using greedy strategy.

One of the main difficulties in analyzing approximation algorithms is to get a handle on the optimum solution. Normally, we do not know the optimum solution of a generic instance of our problem – if we did we would have just output it to begin with. So, instead we use *properties* of the optimum solution to compare its cost with the cost of the solution of our algorithm.

1 Vertex Cover

In the vertex cover problem, we are given an undirected graph. The goal is to find a set of vertices of smallest size, such that every edge of the graph touches one of the vertices in the set.

One option for a greedy algorithm is to pick the vertex of maximum degree, add it to the cover, delete its edges, and repeat. We call this algorithm Greedy1.

Consider the following example: a graph with a set of n vertices A, and sets of vertices $B_1, B_2, \ldots, B_n$, where $|B_i| = \lfloor n/i \rfloor$. Each vertex of A is connected to exactly one vertex of B_i, for all i. The edges into B_i are uniformly distributed so that the degree of a vertex in B_i is approximately i. Then, note that the degree of the vertex in B_n is n and so are the degree of vertices in part A. So Greedy1 may pick the vertex in B_n first. After deleting the edges of the vertex in B_n, the degre of vertices in A goes down to $n-1$, but the degree of the vertex in B_{n-1} is n. So Greedy1 picks B_{n-1}. This keeps happenning until Greedy1 picks B_1. In this way, it picks a cover of size roughly $n + n/2 + n/3 + \ldots$ which is of size $\Omega(n \log n)$, while the set A is a vertex cover of size n. Therefore the approximation ratio of the Greedy1 can be as large as

$$\frac{n \log n}{n} = \log n$$

in the worst case.

Instead, there is a much simpler algorithm that gives an approximation algorithm with approximation ratio 2 for the vertex cover problem.

Input: An undirected graph G
Result: A vertex cover.
Let $T = \emptyset$;
while *Some edge* $\{u, v\}$ *of* G *is uncovered* **do**
  Add both u and v to T;
end
Output T.

Algorithm 1: Greedy2

The analysis is very simple.

Claim 1. *The approximation ratio of Greedy2 is at most 2.*

Proof Let $\{e_1, \ldots, e_k\}$ be the edges chosen by Greedy2 algorithm. By design, T is the set of endpoints of these edges, i.e., $|T| = 2k$ and no two of these edges touch each other.

To prove the claim, it is enough to show that $OPT \geq k$. Consider the graph H only made up of these k edges (so H has $2k$ vertices and k edges). The optimum solution is also a vertex cover for H. But it must have one endpoint of each of the edges, $e_1, \ldots, e_k$ because these edges do not share endpoints. So, Greedy2 has approximation ratio at most 2 in the worst case. ■