CMSC 351 - Introduction to Algorithms
Spring 2012
Lecture 20

Instructor: MohammadTaghi Hajiaghayi
Scribe: Rajesh Chitnis

1 Introduction

In this lecture we will look at Bellman-Ford algorithm for Single-Source Shortest Paths problem and Floyd-Warshall algorithm for All-Pairs Shortest Paths problem.

2 Bellman-Ford algorithm for Single-Source Shortest Paths

Djikstra's algorithm is good if there is no edge of negative length (check this as an exercise!), but the Bellman-Ford works as long as there is no "negative cycle", i.e., a cycle whose edges sum to a negative value and it can even detect such cycles.

Algorithm 1 Bellman-Ford(G,v)

$\mathsf{SP}[v] = 0$;
$\mathsf{SP}[w]=\infty$ for $w \neq v$;
for $i = 1$ to $|V| - 1$ **do**
  for each $(u, w) \in E$ **do**
    if $\mathsf{SP}[u] + \mathsf{length}(u, w) < \mathsf{SP}[w]$ **then**
      $\mathsf{SP}[w] := \mathsf{SP}[u] + \mathsf{length}(u, w)$;

Time Complexity: Clearly it is $O(|V| \cdot |E|)$ time which is worse than Djikstra's $O\Big((|V| + |E|) \log |V|\Big)$.

Proof of Correctness: This can be shown using induction.

Induction Hypothesis
If there is a path from v to u with at most i edges, then SP[u] is at most the length of the shortest path from v to u with at most i edges, where i is the number of repetitions.

Proof: Consider the shortest path P from v to u with at most i edges. Let w be the last vertex before u on this path. Then the part of P from v to w is a shortest path from v to w with at most i-1 edges and by induction SP[w] after i iterations is at most the length of this path. This SP[w]+ length(w,u) is at most the length of P and we find it in the i-th iteration. ■

Now if there is no negative cycle, the length of any shortest path in terms of its edges is at most $|V| - 1$ and thus we find it by Induction Hypothesis for $i = |V| - 1$. The Induction Hypothesis itself is a good property of this algorithm, which has applications in routing protocols as well.

3 Floyd-Warshall Algorithm for All-Pairs Shortest Paths

The Problem: Given a weighted graph $G = (V, E)$ with non-negative edge lengths, find the minimum length paths between all pairs of vertices. Of course we can run Djikstra's algorithm for $|V|$ times to get a total time of $O\Big(|V|(|V| + |E|)\log|V|\Big) = O(|V| \cdot |E| \log |V|)$ which is good for sparse graphs but not the best for dense graphs.

Algorithm 2 Floyd-Warshall(G)

```
for m = 1 to n do
  for x = 1 to n do
    for y = 1 to n do
      if weight[x, m] + weight[m, y] < weight[x, y] then
        weight[x, y] := weight[x, m] + weight[m, y];
```

Time Complexity: Very simple algorithm to implement in $O(|V|^3)$ time.
Proof of Correctness: This can be shown using induction.

Induction Hypothesis
We know the lengths of the shortest paths between all pairs of vertices such that only k-paths, i..e, except endpoints, the highest-labeled vertex on the path is labeled k, for some $k \leq m$, are considered. In i-ith iteration of the loop we computed all these.

Proof: For m=1, the basis is correct since we have only direct edges as paths. For general m, the shortest path between any pair can have v_m at most one and then the paths to and from v_m are k-paths, for $k \leq m - 1$. Since we sum the paths to and from v_m and compare it with the best path found so far in the algorithm, we are done.

■

As you saw in Bellman-Ford and Floyd-Warshall, the whole idea is to get the Induction Hypothesis correct and the rest is then trivial.

References

[1] Udi Manber, *Introduction to Algorithms - A Creative Approach*