Graph Theory: Bellman-Ford

David Jacobo
jguillen@cimat.mx

November 20, 2015

Definition

$$G = (V, E)$$

Discussion

Why aren't we able to solve **Single Source Shortest Paths (SSSP)** correctly with Dijkstra's algorithm under the presence of negative edges?

Outline

1. SSSP w/ negative weights
    - Intuition
    - Algorithm
    - Time/Memory complexity
    - Tips and tricks

Intuition

Given a **weighted graph** and an initial vertex, **calculate the shortest paths to all the other vertices (SSSP)**. Report negative cycles if present and decide the vertices affected by those cycles

Algorithm

Algorithm 1 Set initial distances

```
for $u \in V$ do
    $dist[u] = INF$
    $parent[u] = INV$
end for
$dist[source] = 0$
```

Here INF is a pretty big value, how to choose it? do the math and watch for the largest possible value on your graph. INV is an INValid value, -1 on my code since we know our vertices are given in the range $[0, V-1]$

Algorithm

Algorithm 2 Compute distances

```
for i=0:v-1 do
    for $u \forall V$ do
        for $n : local - neighbours[u]$ do
            $d = dist[u] + n_w$
            if $d < dist[n_y]$ then
                $dist[n_y] = d$
                $parent[n_y] = u$
            end if
        end for
    end for
end for
```

Algorithm

Algorithm 3 Bellman-Ford

```
Set initial distances
Compute distances
Call compute distances again, if a single value is updated
  then we have a negative cycle
Detect vertices which distance will keep decreasing forever
  :′ ( if needed
```

Time complexity: We are running the whole graph (V+E), V-1 times, plus a single run to watch for negative cycles, thus:

$$O(VE)$$

Memory complexity: We need to keep track of the distances and parents per each vertex, so a linear amount of memory is needed with respect to $(w.r.t.)$ V:

$$O(V)$$

This memory analysis doesn't take into account the space to save the graph (that's up to the problem requirement).

Keep in mind...

- Use Bellman-Ford only if the graph is known to have negative edges.
- Remember this algorithm is for a Single Source, we will address the All Pairs Shortest Paths (APSP) later on.
- Relax is again the key component.
- Remember that the V-1 iterations are required because of the case when the given graph is a directed list.

Q & A

References

- Competitive Programming site
- Algorists’ repository