Number of paths between two vertices with length K

Marlon Mendes

1 Problem statement

Given a directed graph $G = (V, E)$, calculate the number of distinct paths with length k between every vertex u to every vertex v such that $u, v \in V$.

2 Dynamic programming solution

Let $f(s, v, k)$ be the number of distinct paths from s to v (this is the solution to the problem) using exactly k edges.

$$f(s,v,k) = \begin{cases} 1 & \text{if } v = s, k = 0 \\ 0 & \text{if } v \neq s, k = 0 \\ \sum\limits^{(u,v)\in E} f(s,u,k-1) & \text{otherwise} \end{cases} \tag{1}$$

This is $O(V^3K)$ in time and $O(V^2K)$ in space. If K is too big (like $K = 10^{18}$) that solution is not fast enough.

3 Speeding up the dynamic programming solution

The above formula for f is clearly a linear recurrence so we can use matrix exponentiation to calculate f. By linear recurrence we're saying that f is a linear combination of other terms of f, for example g(3) = 2 * g(1) + 1 * g(2). Let's represent the graph as an adjacency matrix $M_{n,m}$, such that $M_{i,j} = 1$ if $(i, j) \in E$, and $M_{i,j} = 0$ otherwise.

We want to calculate f(i, j, k) for every i and j so it's a good a idea to start with a matrix A (|V| x |V|). Suppose that that $A[i][j] = f(i, j, k)$, can we multiply that matrix by something and get a new matrix B such that $B[i][j] = f(i, j, k+1)$?

If we look at the formula (1) we now that if B[i][j] = f(i, j, k + 1), then B[i][j] = $\sum^{(u,j) in E} A[i][u]$.

That means that the solution to the (k + 1) problem uses the solution of the k problem, and we add the solution to all vertices of the k problem, as long as there is an edge between u and j, otherwise we can't use vertex u to reach vertex j. Since we're adding the whole row i from matrix A, we could just multiply matrix A by a matrix C such that every element in C is 1, so B = AC, and B[i][j] would be $\sum^{(u,j) in E} A[i][u]$. That would be wrong for the sole reason that we may not respect the condition (u, j) is in E. That is, if the edge (u, j) doesn't exist, then we shouldn't add $A[i][u]$ to the solution. So if Since if $(u, j) \in E$, then we add $A[i][u]$ to the solution, otherwise we don't. So our new matrix C should do that: multiply by 1 terms $A[i][u]$, if $(u, j) \in E$ and multiply by 0 terms $A[i][u]$, if $(u, j) \notin$. This is exactly the adjacency matrix M, so C = M, because if there's no edge (u, v) then M[u][v] = 0, and M[u][v] = 1 otherwise. We can use that same logic to prove by induction that $M^k[i][j]$ = f(i, j, k). There are two possible base cases (and we're already covered the inductive step), case K = 0 and K = 1, however K = 0 is enough.

<u>Case K = 0</u>

We can define M^0 as the identity matrix, so there will be exactly one way to go to u from u using no edges.

<u>Case K = 1</u>

By definition $M^1 = M$, M[i][j] = 1 if there's an edge $(i, j) \in E$, and $M[i][j] = 0$ otherwise.

4 Conclusion

If M is the adjacency matrix for the digraph G, the solution to the problem is M^k, that is, $M^k[i][j]$ = f(i, j, k). We can calculate that using fast exponentiation, making it $O(V^3 log(K))$ in time and $O(V^2)$ in memory.

5 Problems to practice

- UVa 11486 - Finding Paths in Grid
- Codeforces 821E (http://codeforces.com/contest/821/problem/E)