Algorithmic Thinking
Luay Nakhleh

Paths and Matrices

A Quick Tour of Graphs, Matrix Multiplication/Power, and Problem Reduction

We have seen how to use Algorithm **BFS** to find, in linear time, whether there is a path from a node i to a node j in a graph (directed or undirected). In this handout, we will show how to check if there is a path from i to j for any pair of nodes i and j via matrix operations. In other words, while this is a graph-theoretic problem, we will show that we can solve it using an algorithmic technique called **problem reduction**, or **problem transformation**, where we will reduce the problem to another problem about *matrices*, and solve the matrix problem.

1 Paths in graphs

Recall the definition of a path.

Definition 1 *Let $g = (V, E)$ be an undirected graph. There is a path of length k from v_0 to v_k in g, if there is a sequence of k edges in $e_1 = \{v_0, v_1\}, e_2 = \{v_1, v_e\}, \ldots, e_k = \{v_{k-1}, v_k\}$ all of which are in E.*

The same definition applies to paths in directed graphs with the only difference that for $1 \leq i \leq k$, edge e_i is (v_{i-1}, v_i).

A path is *simple* if it does not contain the same node more than once. A *cycle* is a simple path that begins and ends at the same node. To illustrate, consider the graph g in Fig. 1. In this graph, there is a simple path

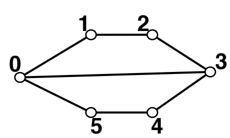

Figure 1: Graph g on 6 nodes.

between nodes 0 and 2 of length 2, which consists of the two edges $\{0, 1\}$ and $\{1, 2\}$. There is also a cycle of length 4 that consists of the edges $\{0, 1\}$, $\{1, 2\}$, $\{2, 3\}$, and $\{0, 3\}$. Notice that there are many other simple paths and cycles of different lengths in this graph.

We say that node j is reachable from node i if there is a path of finite length from i to j. An important result that will be central for the material later in this handout is the following:

Theorem 1 *Let $g = (V, E)$ be a graph with $|V| = n$, and let i and j be two nodes in g. Node j is reachable from node i in g if and only if there is a path of length <u>smaller than n</u> from i to j.*

The implication of this result is that if we cannot find a path of length k for some $k < n$ between two nodes in a graph with n nodes, then we wouldn't find any path of any length between these two nodes.

2 Paths and the adjacency matrix

Consider the question: For a pair of nodes i and j in graph g, is there a path of length 1 from i to j? Of course, the answer is either yes or no. Now, let A_g be the adjacency matrix of graph g and think about the following question: Is there any entry in A_g that helps answer the question? The answer is: $A[i,j]$. If $A[i,j]=1$, then there is an edge between i and j, and the answer to the question is affirmative. If $A[i,j]=0$, then there is no edge between i and j and there can't be a path of length 1 between i and j, so the answer to the question is negative. If you want to verify this, here's the adjacency matrix of g from Fig. 1:

$$A_g = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

Can we generalize this and answer the question: Is there a path from i to j? Notice that from Theorem 1 above, it suffices to check if there is a path of length at most $n-1$ from i to j (where n is the number of nodes in the graph). We now state a theorem without a proof (make sure you understand why it is true).

Theorem 2 *Let g be a graph (directed or undirected), A_g be the adjacency matrix of g, and i and j be two nodes in g. Then, if $A_g^k[i,j] > 0$ for some integer k, then there is a path from i to j.*

Using the two theorems 1 and 2, we can now have an algorithm for testing whether node j is reachable from node i in graph g with n nodes:

1. **For** $k \leftarrow 0$ to $n-1$
   (a) **If** $A_g^k[i,j] \neq 0$ **then**
      i. **Return** *True*;
2. **Return** *False*;

Notice the power of mathematical results and their centrality to algorithm design: Theorem 1 allowed us to bound the loop of Line 1, and Theorem 2 allowed us to use the condition of Line 1(a) to test for connectivity.

It is important to note here that **BFS** is still a more efficient algorithm than this one for checking whether j is reachable from i, but this example is an illustration of how to solve a graph-theoretic problem by transforming it to a matrix problem. In general, this transformation, or reduction, technique is very powerful in computer science: By reducing a problem A to a problem B, we can solve one of the problems using a solution to the other, and we can reason about the computational complexity of one of the problems based on the computational complexity of the other.

3 Multiplication and power of matrices

Let A be an $m \times k$ matrix and B be a $k \times n$ matrix. The product of A and B, denoted by $A \times B$, or AB, is the $m \times n$ matrix C with its $[i,j]$th entry equal to the sum of the products of the corresponding elements from the ith row of A and the jth column of B. In other words,

$$C[i,j] = \sum_{0 \le h \le k-1} A[i,h] \cdot B[h,j]. \tag{1}$$

For example, let

$$A = \begin{bmatrix} 1 & 0 & 4 \\ 2 & 1 & 1 \\ 3 & 1 & 0 \\ 0 & 2 & 2 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 2 & 4 \\ 1 & 1 \\ 3 & 0 \end{bmatrix},$$

and compute the matrix $C = AB$.

Since A is a 4×3 matrix and B is a 3×2 matrix, $C = AB$ is a 4×2 matrix. We now apply Formula (1) to compute the entries of C.

1. $C[0,0] = A[0,0]B[0,0] + A[0,1]B[1,0] + A[0,2]B[2,0] = 1 \times 2 + 0 \times 1 + 4 \times 3 = 14$.
2. $C[0,1] = A[0,0]B[1,0] + A[0,1]B[1,1] + A[0,2]B[2,1] = 1 \times 4 + 0 \times 1 + 4 \times 0 = 4$.
3. $C[1,0] = A[1,0]B[0,0] + A[1,1]B[1,0] + A[1,2]B[2,0] = 2 \times 2 + 1 \times 1 + 1 \times 3 = 8$.
4. $C[1,1] = A[1,0]B[1,0] + A[1,1]B[1,1] + A[1,2]B[2,1] = 2 \times 4 + 1 \times 1 + 1 \times 0 = 9$.
5. $C[2,0] = A[2,0]B[0,0] + A[2,1]B[1,0] + A[2,2]B[2,0] = 3 \times 2 + 1 \times 1 + 0 \times 3 = 7$.
6. $C[2,1] = A[2,0]B[1,0] + A[2,1]B[1,1] + A[2,2]B[2,1] = 3 \times 4 + 1 \times 1 + 0 \times 0 = 13$.
7. $C[3,0] = A[3,0]B[0,0] + A[3,1]B[1,0] + A[3,2]B[2,0] = 0 \times 2 + 2 \times 1 + 2 \times 3 = 8$.
8. $C[3,1] = A[3,0]B[1,0] + A[3,1]B[1,1] + A[3,2]B[2,1] = 0 \times 4 + 2 \times 1 + 2 \times 0 = 2$.

In matrix format, we have

$$C = \begin{bmatrix} 14 & 4 \\ 8 & 9 \\ 7 & 13 \\ 8 & 2 \end{bmatrix}.$$

The *identity matrix* of order n, denoted by I_n, is the $n \times n$ matrix with the entries

$$I_n[i,j] = \begin{cases} 1 & \text{if} \quad i = j \\ 0 & \text{if} \quad i \neq j \end{cases}.$$

An $m \times n$ matrix is called *square* if $n = m$. When A is an $n \times n$ matrix, we define its power as

$$A^0 = I_n \qquad \text{and} \qquad A^r = \underbrace{AAA \cdots A}_{r\ times}.$$

Notice that A must be square for its power to be defined.