



JOINT INSTITUTE
交大密西根学院

VE477 Introduction to Algorithms
Homework 6

Taoyue Xia, 518370910087

2021/11/10

Ex. 1 — Perfect matching in a bipartite graph

1. We denote S_n to be all the permutations of $\{1, 2, \dots, n\}$. Denote one of the permutations as σ , then the signature of σ would be $+1$ if the interchanging of entries to meet the specific permutation can be performed in even number of times. And it will be -1 if interchanging can be performed in odd number of times. Then the product of $a_{1,\sigma_1}, a_{2,\sigma_2}, \dots, a_{n,\sigma_n}$ can be expressed as:

$$\prod_{i=1}^n a_{i,\sigma_i}$$

Therefore, the final expression of calculating the determinant would be:

$$\det(A) = \sum_{\sigma \in S_n} (\text{sgn}(\sigma) \prod_{i=1}^n a_{i,\sigma_i})$$

\Rightarrow Since every vertex will be contained in one edge, it means that there will exist at most one $X_{i,j}$ in each row and column, with others all 0. Therefore, if the determinant of A is identically zero, it means that there exists some row or column with all the elements being 0, so that all the product of permutations will give 0. In this sense, it tells that some node in L or R is not contained in edge in $G.E$. So we have got to the point that G has no perfect matching.

\Leftarrow If G has no perfect matching, it means that at least one vertex in L or R are not contained in $G.E$, which will make one row or one column full of zeros. Therefore, since in all permutations, the product of all a_{i,σ_i} is calculated, with one row or column with all the elements being 0, all the products of all permutations will give the answer 0. Therefore, by adding them together, the determinant will be identically zero. Proof done.

2. To decide whether a bipartite graph has a perfect matching, we just need to first construct an adjacency matrix A of the graph, and set $X_{i,j}$ to be random positive integers. Then, if we can find some row or column full of zeros, it means that the graph does not contain a perfect matching. If we cannot find such row or column, it can be a perfect matching. The algorithm is shown below:

Algorithm 1: Perfect matching decision

Input : A bipartite graph $G = \langle V, E \rangle$, with $V = L \cup R$

Output: True or False

```
/* Denote  $L = \{l_1, l_2, \dots, l_n\}$  and  $R = \{r_1, r_2, \dots, r_n\}$  */
/* Construct the matrix */
1  $A \leftarrow$  a two-dimension array acting as a matrix;
2 for  $i = 1$  to  $n$  do
3   for  $j = 1$  to  $n$  do
4     if  $(l_i, r_j)$  is in  $E$  then
5        $x \leftarrow$  a random positive integer;
6        $A[i][j] \leftarrow x$ ;
7     else
8        $A[i][j] \leftarrow 0$ ;
9     end
10  end
11 end

/* Check empty row or column */
12  $countRow \leftarrow 0$ ;
13  $countColumn \leftarrow 0$ ;
14 for  $i = 1$  to  $n$  do
15   for  $j = 1$  to  $n$  do
16     if  $A[i][j] = 0$  then
17        $countRow \leftarrow countRow + 1$ ;
18     end
19     if  $A[j][i] = 0$  then
20        $countColumn \leftarrow countColumn + 1$ ;
21     end
22   end
23   if  $countRow = n$  or  $countColumn = n$  then
24     return False;
25   end
26    $countRow, countColumn \leftarrow 0$ ;
27 end
28 return True;
```

3. Suppose that there are n vertices in both L and R , then the time complexity of the previous algorithm is $\mathcal{O}(n^2)$. Error will occur when the determinant calculated is 0, but there exists no row or column to be all zero.

4. Using Ford-Fulkerson Algorithm we can simply determine whether it has a perfect matching in $\mathcal{O}(VE)$ time. However, this strategy uses the concept of an adjacency matrix, and by calculating its determinant or checking the rows and columns, we can reach the goal. It is more like a linear algebra solution, which is more graphic.