

求图中任意两顶点的最短路径算法的 C++实现

// edge[n][n]是有 n 个顶点的图的邻接矩阵; A[i][j]是顶点 V_i 至 V_j 的最短路径长度;

// path[i][j]是相应路径上顶点 j 的前一个顶点的序号。

void Graph_Matrix :: AllLength ()

```
{
    int n = graphsize;
    int **path = new int*[n];
    int **A = new int*[n];
    for(int i=0; i<n; i++)
    {
        path[i] = new int[n];
        A[i] = new int[n];
    }
    for( i=0 ; i<n ; i++ )                // 矩阵 A 和 path 初始化
        for(j=0 ; j<n ; j++ )
        {
            A[i][j] = edge[i][j] ;        // 初始化的 A 即  $A^{(0)}$ 
            if ( i != j && A[i][j] < max ) // max 是预定义的常数
                path[i][j] = i ;
            else
                path[i][j] = -1 ;
        }
    for ( k=0 ; k<n ; k++ )                // 从  $A^{(-1)}$ 开始构造  $A^{(n-1)}$ 
        for ( i=0 ; i<n ; i++ )
            if ( i != k )
                for ( j=0 ; j<n ; j++ )
                    if( j!=k && j!=i && A[i][k]<max && A[k][j]<max && A[i][k]+A[k][j]<A[i][j])
                    {
                        A[i][j] = A[i][k] + A[k][j] ;
                        path[i][j] = path[k][j] ;
                    }
    for(i=0; i<n; i++)
    {
        delete[] path[i];
        delete[] A[i];
    }
    delete[] path;
    delete[] A;
}
```