

Типовые алгоритмы на ациклических графах.

Эти алгоритмы являются применением динамического программирования.

a - матрица смежностей с бесконечностями (INFINITY) ориентированного ациклического графа

b - матрица того же размера, что матрица a

<pre>1. Алгоритм построения матрицы кратчайших путей // b - заполнена нулями for(int i=0; i<N; ++i) for(int k=i+1; k<N; ++k) for(int j=k+1; j<N; ++j) if (a[i][j]>a[i][k]+a[k][j]) { a[i][j] = a[i][k]+a[k][j]; b[i][j] = k; } // a - матрица кратчайших расстояний, b - матрица восстановления путей 2. Алгоритм построения матрицы самых длинных (критических) путей // a - матрица смежностей с -INFINITY, b - заполнена нулями for(int i=0; i<N; ++i) for(int k=i+1; k<N; ++k) for(int j=k+1; j<N; ++j) if (a[i][j]<a[i][k]+a[k][j]) { a[i][j] = a[i][k]+a[k][j]; b[i][j] = k; } // a - матрица самых длинных расстояний, b - матрица восстановления путей 3. Алгоритм построения матрицы количеств путей // b – заполнена нулями for(int i=0; i<N; ++i) for(int k=i+1; k<N; ++k) for(int j=k+1; j<N; ++j) if (a[k][j]<INFINITY) b[i][j] += b[i][k]; // b - матрица количества путей</pre>	<pre>4. Алгоритм поиска путей наибольшей пропускной способности // a - содержит пропускные способности, b - заполнена нулями for(int i=0; i<N; ++i) for(int k=i+1; k<N; ++k) for(int j=k+1; j<N; ++j) if (a[i][j]<min(a[i][k],a[k][j])) { a[i][j] = min(a[i][k],a[k][j]); b[i][j] = k; } // a - матрица путей наибольшей пропускной способности, b - матрица восстановления путей 5. Алгоритм поиска самых надежных путей // a - содержит вероятности, b - заполнена нулями for(int i=0; i<N; ++i) for(int k=i+1; k<N; ++k) for(int j=k+1; j<N; ++j) if (a[i][j]<a[i][k]*a[k][j]) { a[i][j] = a[i][k]*a[k][j]; b[i][j] = k; } // a - матрица самых надежных путей, b - матрица восстановления путей</pre>
--	--