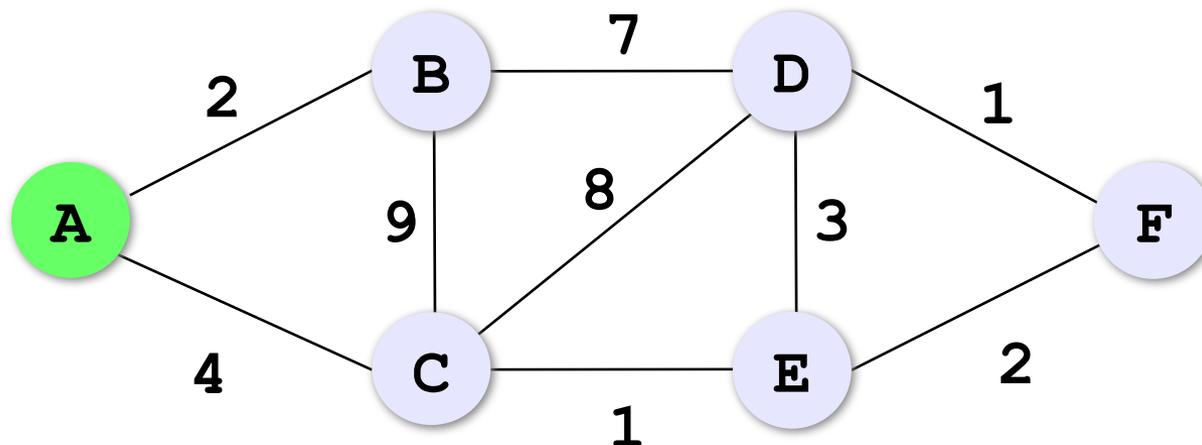


Кратчайший маршрут

Алгоритм Дейкстры (1960):



Э.В.
Дейкстра

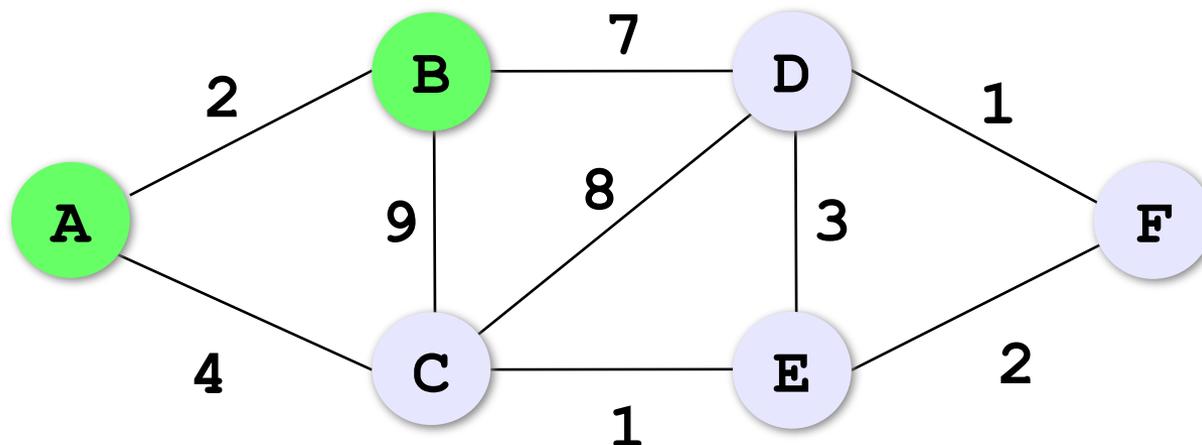
	A	B	C	D	E	F
R	0	2	4	∞	∞	∞
P	x	A	A	A	A	A

кратчайшее расстояние
откуда ехать

ближайшая от A
невыбранная вершина

Кратчайший маршрут

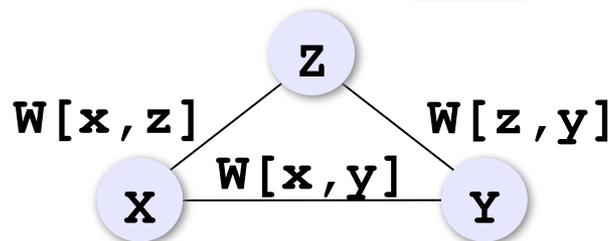
Алгоритм Дейкстры (1960):



Э.В.
Дейкстра

	A	B	C	D	E	F
R	0	2	4	9	∞	∞
P	x	A	A	B	A	A

кратчайшее расстояние
откуда ехать

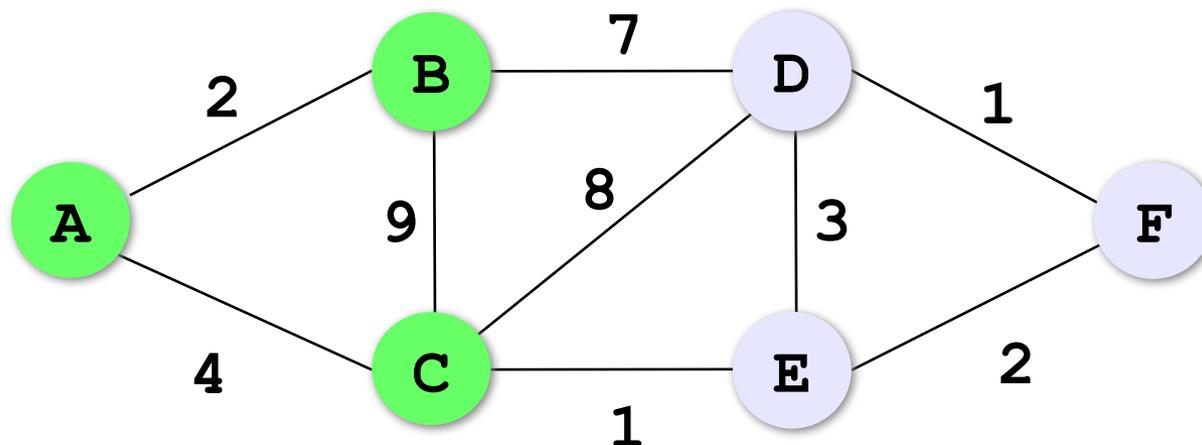


может быть так, что

$$W[x, z] + W[z, y] < W[x, y]$$

Кратчайший маршрут

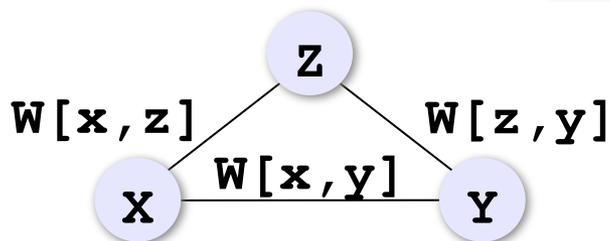
Алгоритм Дейкстры (1960):



Э.В.
Дейкстра

	A	B	C	D	E	F
R	0	2	4	9	5	∞
P	x	A	A	B	C	A

кратчайшее расстояние
откуда ехать

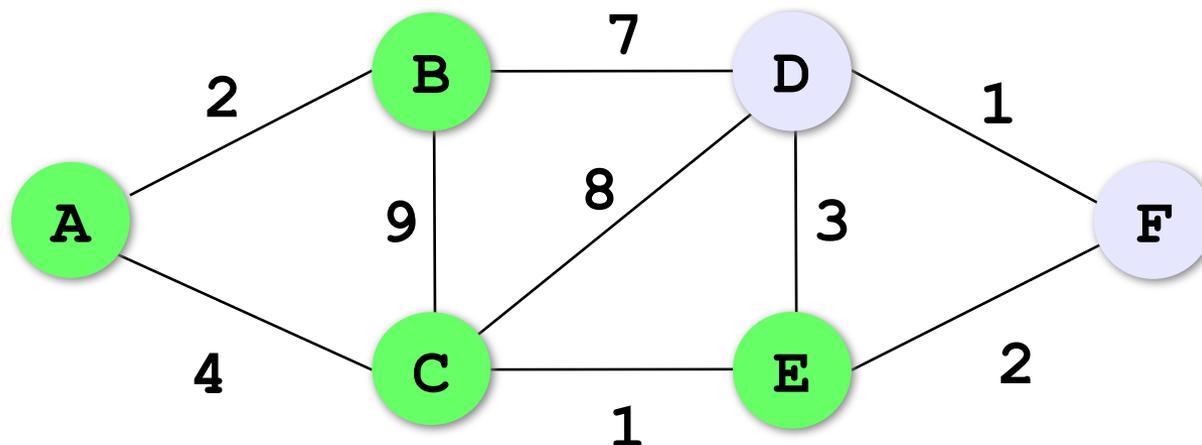


может быть так, что

$$W[x, z] + W[z, y] < W[x, y]$$

Кратчайший маршрут

Алгоритм Дейкстры (1960):



Э.В.
Дейкстра

	A	B	C	D	E	F
R	0	2	4	8	5	7
P	x	A	A	E	C	E

кратчайшее расстояние
откуда ехать



При рассмотрении вершин **F** и **D**
таблица не меняется!

Кратчайший маршрут

	А	В	С	Д	Е	Ф
Р	0	2	4	8	5	7
Р	х	А	А	Е	С	Е

длины кратчайших маршрутов из А в другие вершины

?

Как найти сам маршрут?

	А	В	С	Д	Е	Ф
Р	0	2	4	8	5	7
Р	х	А	А	Е	С	Е

А → С → Е → Ф

Алгоритм Дейкстры

Данные:

```
const int N = 6;  
int W[N][N];    // весовая матрица  
bool active[N]; // вершина не выбрана?  
int R[N], P[N];  
int i, j, min, kMin;
```

Начальные значения (выбор начальной вершины):

```
for ( i = 0; i < N; i++ ) {  
    active[i] = true; // все вершины не выбраны  
    R[i] = W[0][i];  // рёбра из вершины 0  
    P[i] = 0;  
}  
active[0] = false; // вершина уже выбрана  
P[0] = -1;        // это начальная вершина
```

Алгоритм Дейкстры

Основной цикл:

```
for ( i = 0; i < N-1; i++ ) {  
    minDist = 99999;  
    for ( j = 0; j < N; j++ )  
        if ( active[j] && R[j] < minDist ) {  
            minDist = R[j];  
            kMin = j;  
        }  
    active[kMin] = false;  
    for ( j = 0; j < N; j++ )  
        if ( R[kMin] + W[kMin][j] < R[j] ) {  
            R[j] = R[kMin] + W[kMin][j];  
            P[j] = kMin;  
        }  
}
```

выбор следующей
вершины,
ближайшей к A

проверка
маршрутов через
вершину kMin

Алгоритм Дейкстры

Вывод результата (маршрут $0 \rightarrow N-1$):

```

i = N-1;
while ( i != -1 )
{
  cout << i << " ";
  i = P[i]; // к следующей вершине
}

```

для начальной
вершины $P[i] = -1$

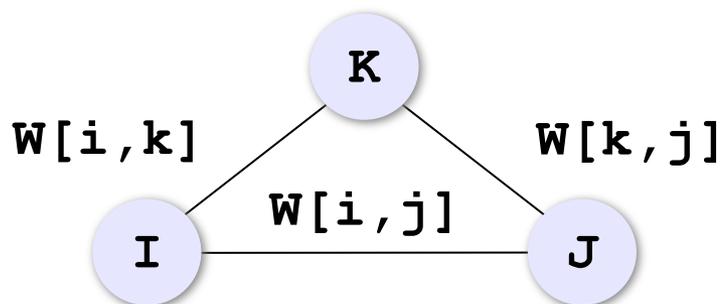
	A	B	C	D	E	F
R	0	2	4	8	5	7
P	x	A	A	E	C	E

A → C → E → F

Алгоритм Флойда

Все кратчайшие пути (из любой вершины в любую):

```
for ( k = 0; k < N; k++ )  
  for ( i = 0; i < N; i++ )  
    for ( j = 0; j < N; j++ )  
      if ( W[i][k] + W[k][j] < W[i][j] )  
        W[i][j] = W[i][k] + W[k][j];
```



Как найти сам маршрут?

Алгоритм Флойда + маршруты

Дополнительная матрица:

```
for ( i = 0; i < N; i++ ) {  
    for ( j = 0; j < N; j++ )  
        P[i][j] = i;  
    P[i][i] = -1;  
}
```

Кратчайшие длины путей и маршруты:

```
for ( k = 0; k < N; k++ )  
    for ( i = 0; i < N; i++ )  
        for ( j = 0; j < N; j++ )  
            if ( W[i][k] + W[k][j] < W[i][j] ) {  
                W[i][j] = W[i][k] + W[k][j];  
                P[i][j] = P[k][j];  
            }
```

Конец фильма

ПОЛЯКОВ Константин Юрьевич

д.т.н., учитель информатики

ГБОУ СОШ № 163, г. Санкт-Петербург

kpolyakov@mail.ru

ЕРЕМИН Евгений Александрович

к.ф.-м.н., доцент кафедры мультимедийной

дидактики и ИТО ПГГПУ, г. Пермь

eremin@pspu.ac.ru