

1. Введение

- 1) Текстовая формулировка задачи
- 2) Пример кода, решающего данную задачу
- 3) Скриншот программы

2. Ход работы

2.1. Код приложения

Кратчайшие пути на графе

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
#define SIZE 6
int main()
{
    int a[SIZE][SIZE];
    int d[SIZE];
    int v[SIZE];
    int temp, minindex, min;
    int begin_index = 0;
    system("chcp 1251");
    system("cls");
    for (int i = 0; i < SIZE; i++)
    {
        a[i][i] = 0;
        for (int j = i + 1; j < SIZE; j++) {
            printf("Введите расстояние %d - %d: ", i + 1, j + 1);
            scanf("%d", &temp);
            a[i][j] = temp;
            a[j][i] = temp;
        }
    }
    for (int i = 0; i < SIZE; i++)
    {
        for (int j = 0; j < SIZE; j++)
            printf("%5d ", a[i][j]);
        printf("\n");
    }
    for (int i = 0; i < SIZE; i++)
    {
```

```

    d[i] = 10000;
    v[i] = 1;
}
d[begin_index] = 0;
do {
    minindex = 10000;
    min = 10000;
    for (int i = 0; i<SIZE; i++)
    {
        if ((v[i] == 1) && (d[i]<min))
        {
            min = d[i];
            minindex = i;
        }
    }
    if (minindex != 10000)
    {
        for (int i = 0; i<SIZE; i++)
        {
            if (a[minindex][i] > 0)
            {
                temp = min + a[minindex][i];
                if (temp < d[i])
                {
                    d[i] = temp;
                }
            }
        }
        v[minindex] = 0;
    }
} while (minindex < 10000);
printf("\nКратчайшие расстояния до вершин: \n");
for (int i = 0; i<SIZE; i++)
    printf("%5d ", d[i]);

int ver[SIZE];
int end = 4;
ver[0] = end + 1;
int k = 1;
int weight = d[end];

while (end != begin_index)
{
    for (int i = 0; i<SIZE; i++)

```

```

    if (a[i][end] != 0)
    {
        int temp = weight - a[i][end];
        if (temp == d[i])
        {
            weight = temp;
            end = i;
            ver[k] = i + 1;
            k++;
        }
    }
}
printf("\nВывод кратчайшего пути\n");
for (int i = k - 1; i >= 0; i--)
    printf("%3d ", ver[i]);
getchar(); getchar();
return 0;
}

```

2.2. формулы

$S[i]:=1$; $C[i]:=0$ $B[j] \leq B[k]$ Затем выполняются следующие операции: $S[j]:=1$; если $B[k] > B[j] + A[j,k]$, то $(B[k]:=B[j] + A[j,k]; C[k]:=j)$

3. Пример скриньшота программы

Кратчайшие пути на графе

```
sh: 1: chcp: not found
sh: 1: cls: not found
Введите расстоян 1 - 2: 2
Введите расстоян 1 - 3: 3
Введите расстоян 1 - 4: 4
Введите расстоян 1 - 5: 2
Введите расстоян 1 - 6: 3
Введите расстоян 2 - 3: 2
Введите расстоян 2 - 4: 3
Введите расстоян 2 - 5: 3
Введите расстоян 2 - 6: 5
Введите расстоян 3 - 4: 3
Введите расстоян 3 - 5: 3
Введите расстоян 3 - 6: 5
Введите расстоян 4 - 5: 4
Введите расстоян 4 - 6: 3
Введите расстоян 5 - 6: 4
  0   2   3   4   2   3
  2   0   2   3   3   5
  3   2   0   3   3   5
  4   3   3   0   4   3
  2   3   3   4   0   4
  3   5   5   3   4   0

Кратчайшие расстояния до вершин:
  0   2   3   4   2   3
Вывод кратчайшего пути
  1   5
```

Рис. 1. скриньшот программы

```

1 #define _CRT_SECURE_NO_WARNINGS
2 #include <stdio.h>
3 #include <stdlib.h>
4 #define SIZE 6
5 int main()
6 {
7     int a[SIZE][SIZE]; // матрица связей
8     int d[SIZE]; // минимальное расстояние
9     int v[SIZE]; // посещенные вершины
10    int temp, minindex, min;
11    int begin_index = 0;
12    system("chcp 1251");
13    system("cls");
14    // Инициализация матрицы связей
15    for (int i = 0; i < SIZE; i++)
16    {
17        a[i][i] = 0;
18        for (int j = i + 1; j < SIZE; j++) {
19            printf("Введите расстояние %d - %d: ", i + 1, j + 1);
20            scanf("%d", &temp);
21            a[i][j] = temp;
22            a[j][i] = temp;
23        }
24    }
25    // Вывод матрицы связей
26    for (int i = 0; i < SIZE; i++)
27    {
28        for (int j = 0; j < SIZE; j++)
29            printf("%5d ", a[i][j]);
30        printf("\n");
31    }
32    // Инициализация вершин и расстояний
33    for (int i = 0; i < SIZE; i++)
34    {
35        d[i] = 10000;
36        v[i] = 1;
37    }
38    d[begin_index] = 0;
39    // Шаг алгоритма
40    do {
41        minindex = 10000;
42        min = 10000;
43        for (int i = 0; i < SIZE; i++)

```

Рис. 2. скриншот программы

```

44- { // Если вершину ещё не обошли и вес меньше min
45-   if ((v[i] == 1) && (d[i] < min))
46-   { // Переприсваиваем значения
47-     min = d[i];
48-     minindex = i;
49-   }
50- }
51- // Добавляем найденный минимальный вес
52- // к текущему весу вершины
53- // и сравниваем с текущим минимальным весом вершины
54- if (minindex != 10000)
55- {
56-   for (int i = 0; i < SIZE; i++)
57-   {
58-     if (a[minindex][i] > 0)
59-     {
60-       temp = min + a[minindex][i];
61-       if (temp < d[i])
62-       {
63-         d[i] = temp;
64-       }
65-     }
66-   }
67-   v[minindex] = 0;
68- }
69- } while (minindex < 10000);
70- // Вывод кратчайших расстояний до вершин
71- printf("\nКратчайшие расстояния до вершин: \n");
72- for (int i = 0; i < SIZE; i++)
73-   printf("%5d ", d[i]);
74-
75- // Восстановление пути
76- int ver[SIZE]; // массив посещенных вершин
77- int end = 4; // индекс конечной вершины = 5 - 1
78- ver[0] = end + 1; // начальный элемент - конечная вершина
79- int k = 1; // индекс предыдущей вершины
80- int weight = d[end]; // вес конечной вершины
81-
82- while (end != begin_index) // пока не дошли до начальной вершины
83- {
84-   for (int i = 0; i < SIZE; i++) // просматриваем все вершины

```

Рис. 3. скриншот программы

```

83 ~ {
84     for (int i = 0; i < SIZE; i++) // просматриваем все вершины
85         if (a[i][end] != 0) // если связь есть
86         {
87             int temp = weight + a[i][end]; // определяем вес пути из предыдущей вершины
88             if (temp == d[i]) // если вес совпал с рассчитанным
89                 // значит из этой вершины и был переход
90                 weight = temp; // сохраняем новый вес
91                 end = i; // сохраняем предыдущую вершину
92                 ver[k] = i + 1; // и записываем ее в массив
93                 k++;
94         }
95     }
96 }
97 // Вывод пути (начальная вершина оказалась в конце массива из k элементов)
98 printf("\nВывод кратчайшего пути\n");
99 for (int i = k - 1; i >= 0; i--)
100     printf("%3d ", ver[i]);
101 getchar(); getchar();
102 return 0;
103 }

```

Рис. 4. скриншот программы

4. библиографические ссылки

Для изучения «внутренностей» \TeX необходимо изучить [1], а для использования \LaTeX лучше почитать [2, 3].

Список литературы

- [1] Кнут Д.Э. Всё про \TeX . — Москва: Изд. Вильямс, 2003 г. 550 с.
- [2] Львовский С.М. Набор и верстка в системе \LaTeX . — 3-е издание, исправленное и дополненное, 2003 г.
- [3] Воронцов К.В. \LaTeX в примерах. 2005 г.