

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  #include <iostream>
2
3  using namespace std;
4
5  void quick_sort(int arr[], int left, int right) {
6      int i = left;
7      int j = right;
8      int tmp;
9      int pivot = arr[(left + right) / 2];
10     while (i <= j) {
11         while (arr[i] < pivot)
12             i++;
13         while (arr[j] > pivot)
14             j--;
15         if (i <= j) {
16             tmp = arr[i];
17             arr[i] = arr[j];
18             arr[j] = tmp;
19             i++;
20             j--;
21         }
22     }
23     if (left < j)
24         quick_sort(arr, left, j);
25     if (i < right)
26         quick_sort(arr, i, right);
27 }
28
29 void merge(int arr[], int p, int q, int r) {
30     int n1 = q - p + 1;
31     int n2 = r - q;
32     int L[n1 + 1], R[n2 + 1];
33     for (int i = 0; i < n1; i++)
34         L[i] = arr[p + i];
35     for (int j = 0; j < n2; j++)
36         R[j] = arr[q + 1 + j];
37     L[n1] = 1000000; // sentinel value
38     R[n2] = 1000000; // sentinel value
39     int i = 0, j = 0;
40     for (int k = p; k <= r; k++) {

```

Рис. 2. скриньшот программы

```

41-     if (L[i] <= R[j]) {
42-         arr[k] = L[i];
43-         i++;
44-     } else {
45-         arr[k] = R[j];
46-         j++;
47-     }
48- }
49- }
50-
51- void merge_sort(int arr[], int p, int r) {
52-     if (p < r) {
53-         int q = (p + r) / 2;
54-         merge_sort(arr, p, q);
55-         merge_sort(arr, q + 1, r);
56-         merge(arr, p, q, r);
57-     }
58- }
59-
60- int main() {
61-     int arr1[] = {5, 3, 6, 2, 10};
62-     int arr2[] = {15, 23, 4, 17, 8};
63- }
64- cout << "Quick sort:\n";
65- quick_sort
66-

```

Рис. 3. 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.