Учреждение образования

«Белорусский государственный технологический университет»

**Отчет по лабораторным работам по дисциплине**

**“Математическое программирование”**

Выполнил:

Студент 2 курса 4 группы ФИТ

Корнелюк Валентин Владимирович

2023 г.

**Лабораторная работа №1**

**Вспомогательные функции**

**Цель работы:** приобретение навыков составления и отладки программ с использованием пользовательских функций для замера продолжительности процесса вычисления.

**Задание 1**

Разработайте три функции (start, dget и iget), используя следующие спецификации. Реализация представлена на рисунке 1.

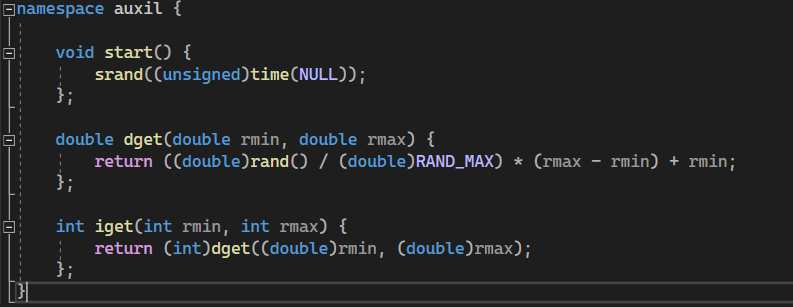


Рисунок 1 – Реализация функций

**Задание 2**

1.Реализовать пример 2.

2.Для проверки работоспособности разработанных функций и приобретения навыков замера продолжительности процесса вычисления реализуйте программу, приведенную в примере 2.

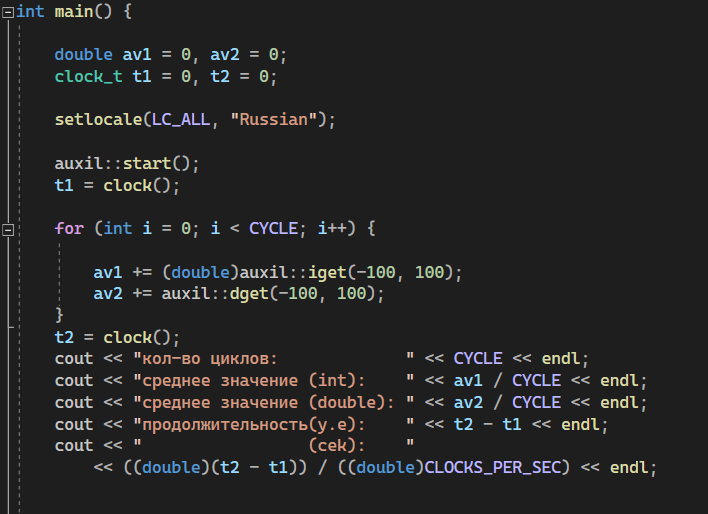


Рисунок 2 – вызов функции и измерение времени выполнения

**Задание 3**

Проведите необходимые эксперименты (разработать кодом) и постройте график зависимости (Excel) продолжительности процесса вычисления от количества циклов в примере 2.

Примечание: продолжительность вычисления измерять в условных единицах процессорного времени (функция clock). График представлен на рисунке 3

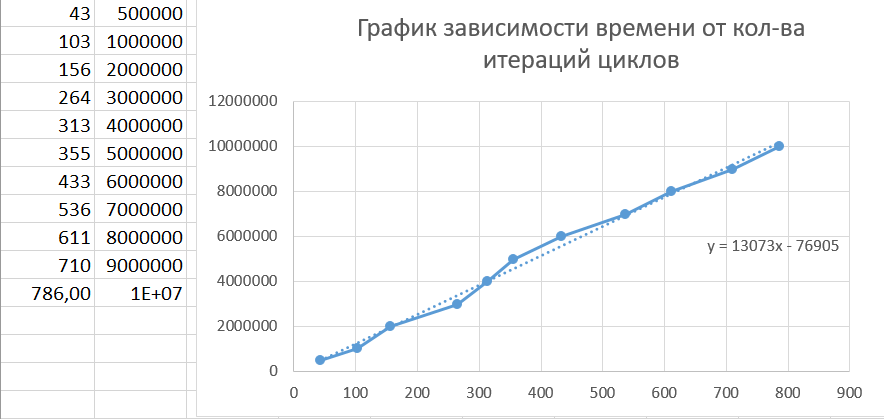


Рисунок 3 – график зависимости

Реализация Факториала в коде демонстрируется на рис. 4

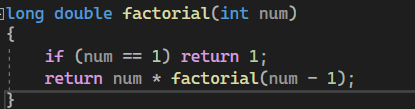


Рисунок 4 – функция для нахождения факториала

Вызов и подсчет времени для реализации факториала представлен на рисунке 5:

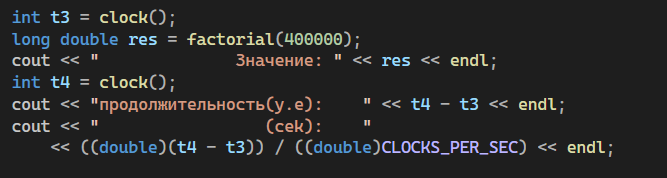


Рисунок 5 – вызов функции для подсчёта факториала

График зависимости:

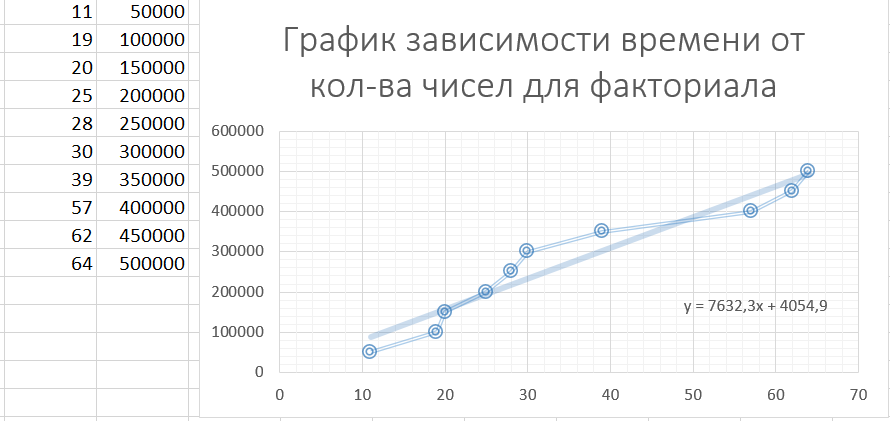


Рисунок 6 – график зависимости

**Вывод**: по полученным измерениям, можем сделать следующий вывод, что время выполнения программы линейно зависит от количества циклов или значения факториала.

**Лабораторная работы №2**

**Комбинаторные алгоритмы решения оптимизационных задач**

**ЦЕЛЬ РАБОТЫ:** приобрести навыки разработки генераторов подмножеств, перестановок, сочетаний и размещений на С++; научиться применять разработанные генераторы для решения задач о рюкзаке (упрощенную, коммивояжера, об оптимальной загрузке судна и об оптимальной загрузке судна с центровкой.

**Задание 1.** Разобрать и разработать генератор подмножеств заданного множества.

Ниже на рисунках представлены заголовочные файлы и файлы cpp с реализацией поставленной задачи.

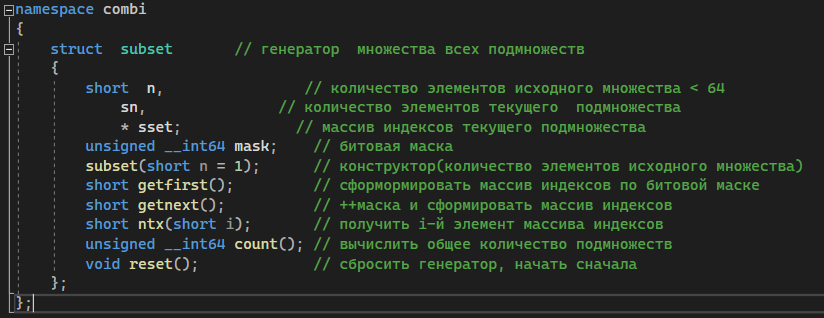


Рисунок 1 – Combi.h

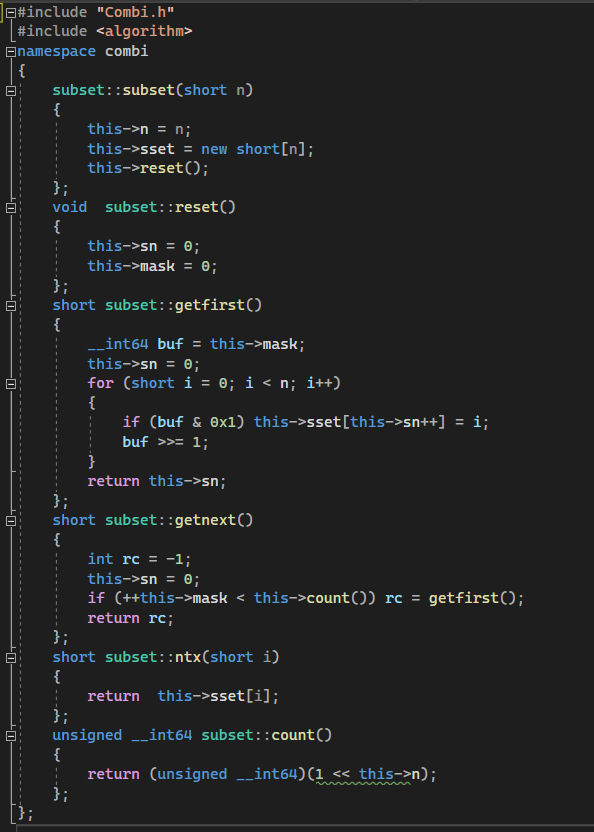


Рисунок 2 - Combi.cpp

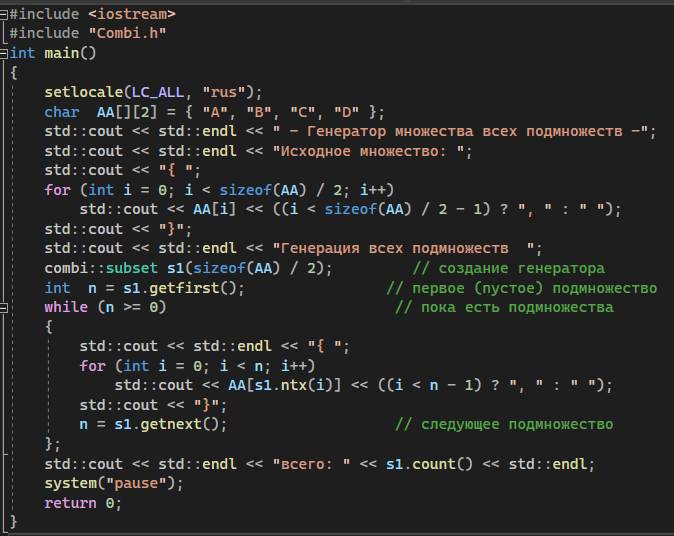


Рисунок 3 - main.cpp

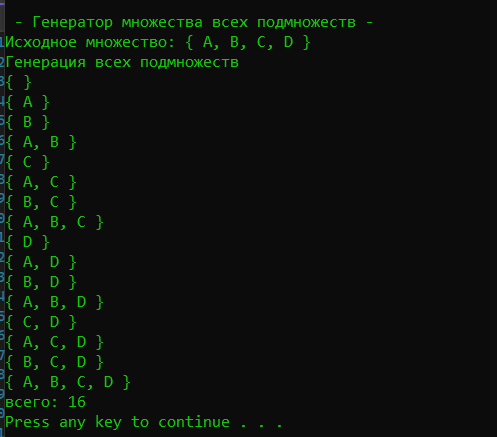


Рисунок 4 – Результат выполнения

**Задание 2.** Разобрать и разработать генератор сочетаний.

Ниже на рисунках представлены заголовочные файлы и файлы cpp с реализацией поставленной задачи.

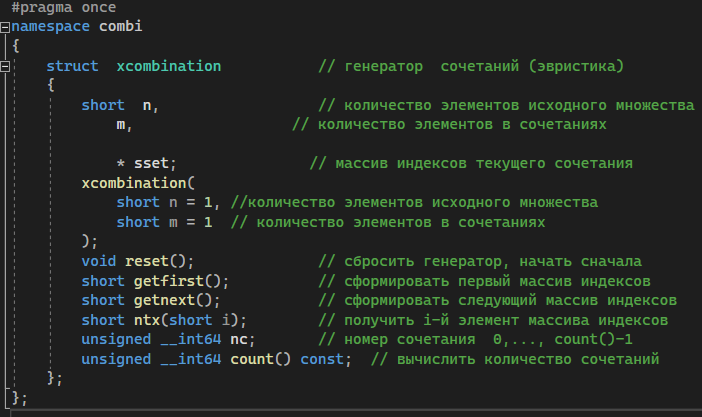


Рисунок 5 - Combi.h

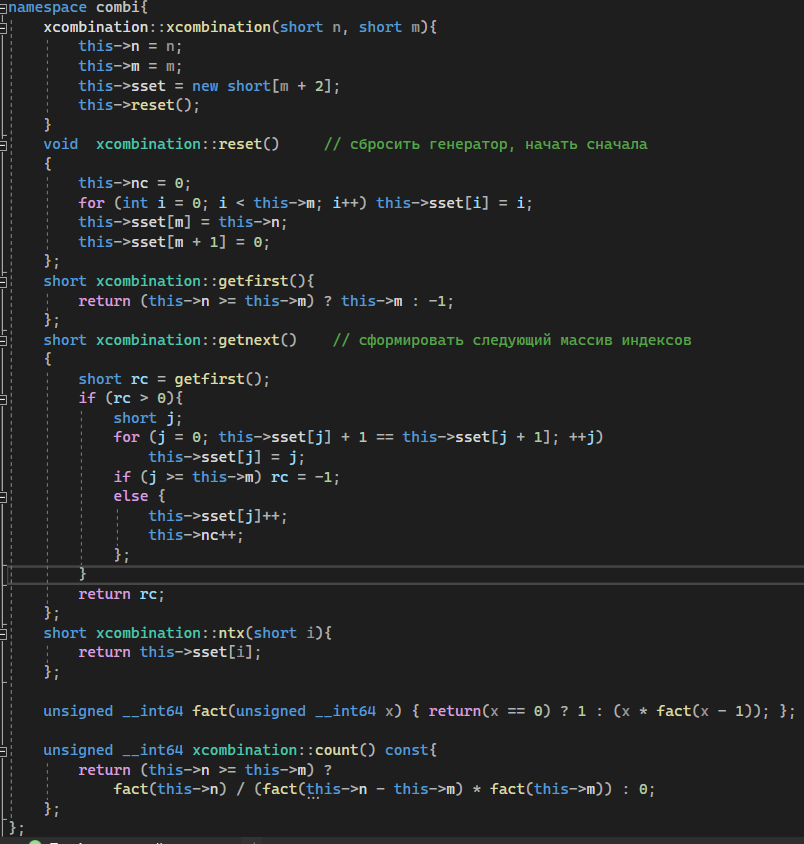


Рисунок 6 - Combi.cpp

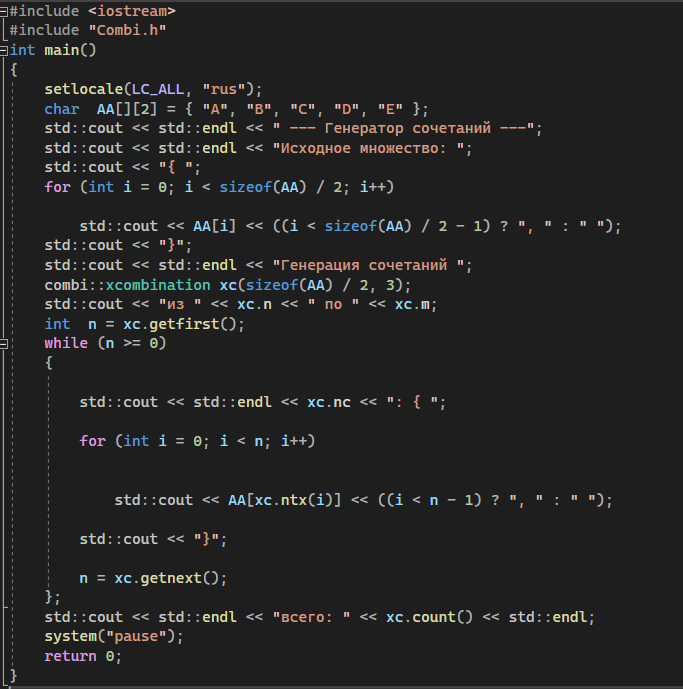


Рисунок 7 - main.cpp

Результат выполнения представлен на рисунке 8.

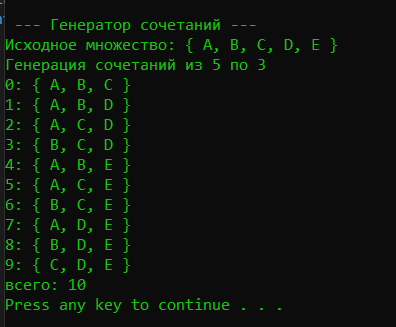


Рисунок 8 – результат выполнения программы

**Задание 3.** Разобрать и разработать генератор перестановок.

Ниже на рисунках представлены заголовочные файлы и файлы cpp с реализацией поставленной задачи.

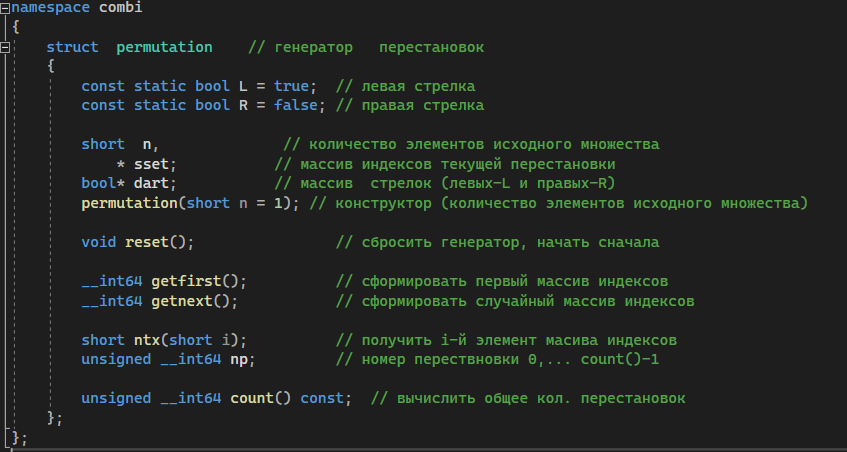


Рисунок 9 – Combi.h

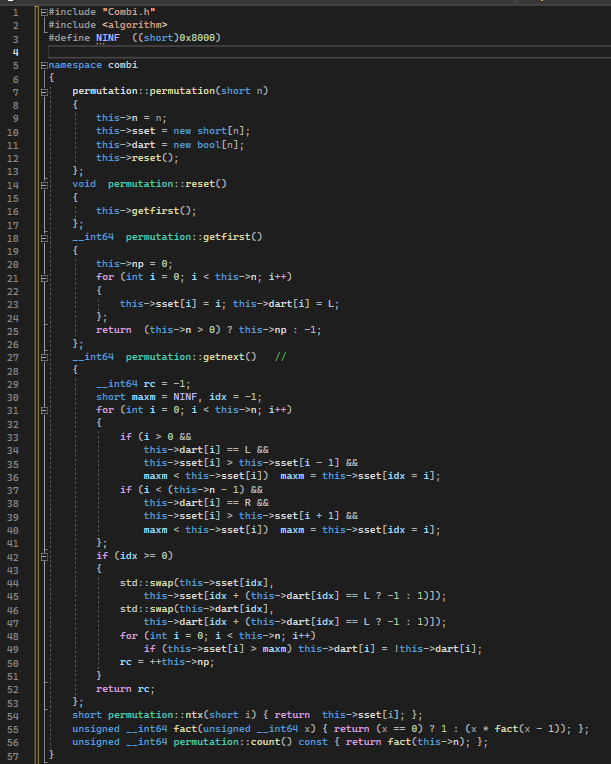


Рисунок 10 – Combi.cpp

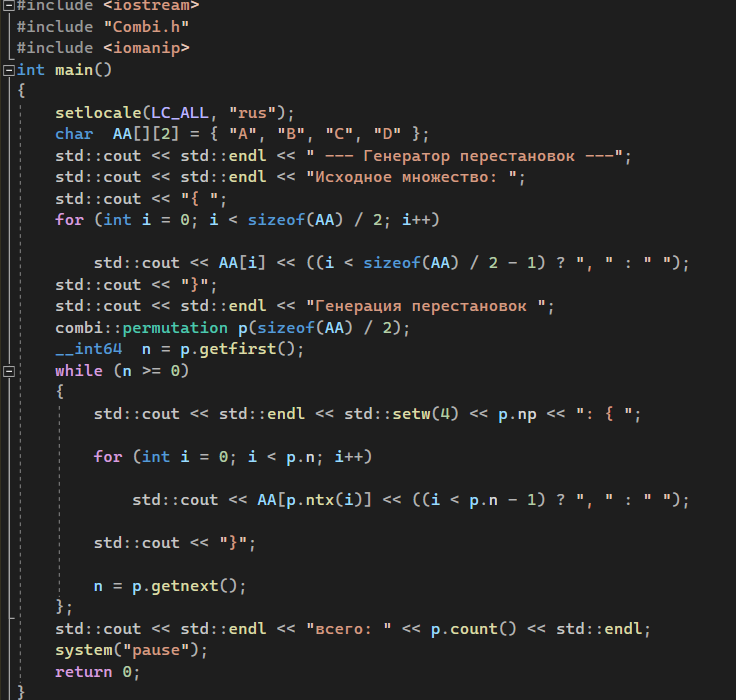


Рисунок 11 – main.cpp

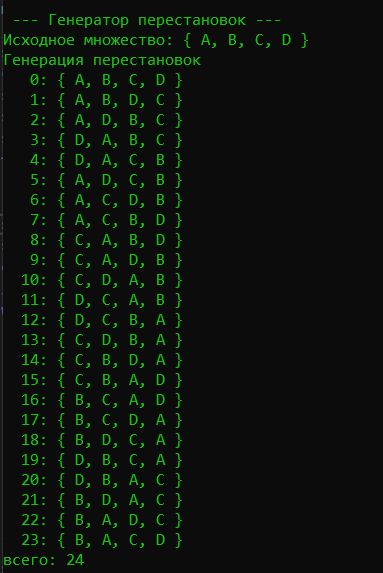


Рисунок 12 – результат выполнения программы

**Задание 4.** Разобрать и разработать генератор размещений.

Ниже на рисунках представлены заголовочные файлы и файлы cpp с реализацией поставленной задачи.

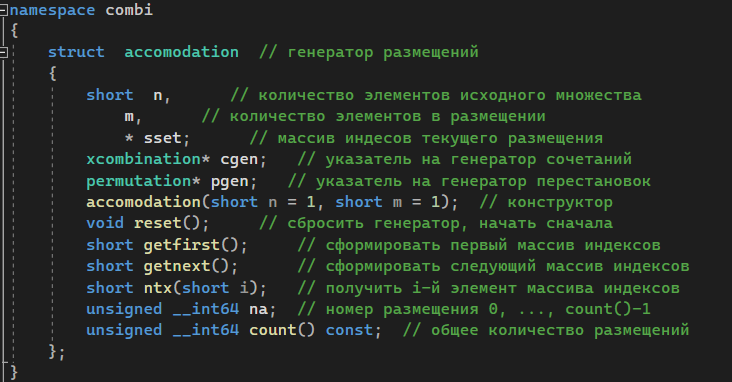


Рисунок 13 – Combi.h



Рисунок 14 – Combi.cpp

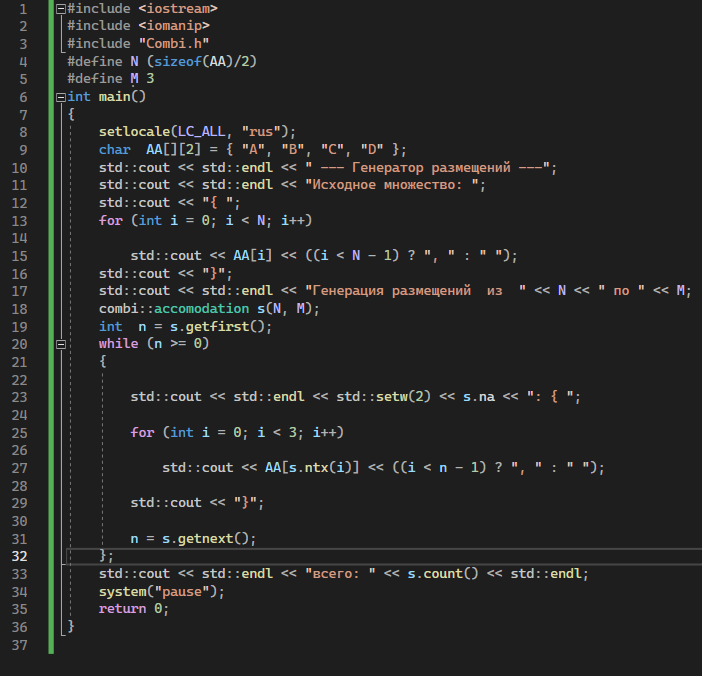


Рисунок 15 – main.cpp

Результат выполнения программы представлен на рисунке 16.

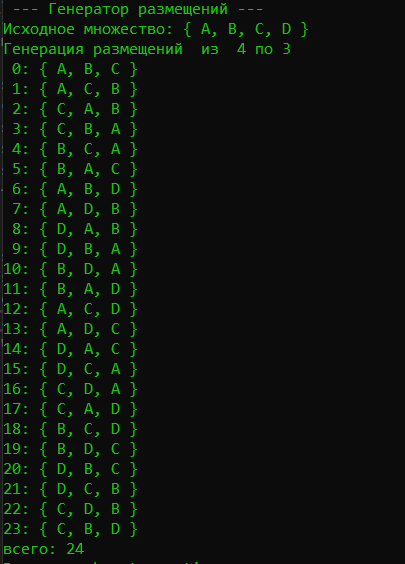


Рисунок 16 - результат выполнения программы

**Задание 5.** Решить в соответствии с вариантом задачу и результат занести в отчет. Решить задачу коммивояжера (расстояния сгенерировать случайным образом: 10 городов, расстояния 10 – 300 км, 3 расстояния между городами задать бесконечными);



Рисунок 17 – схема решения задачи

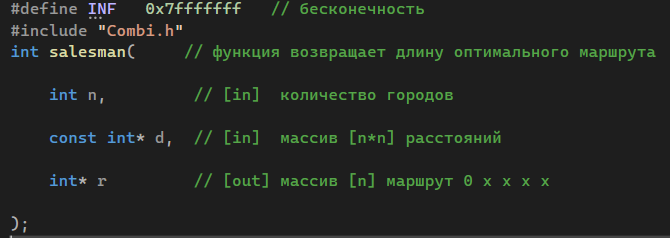


Рисунок 17 – Salesman.h

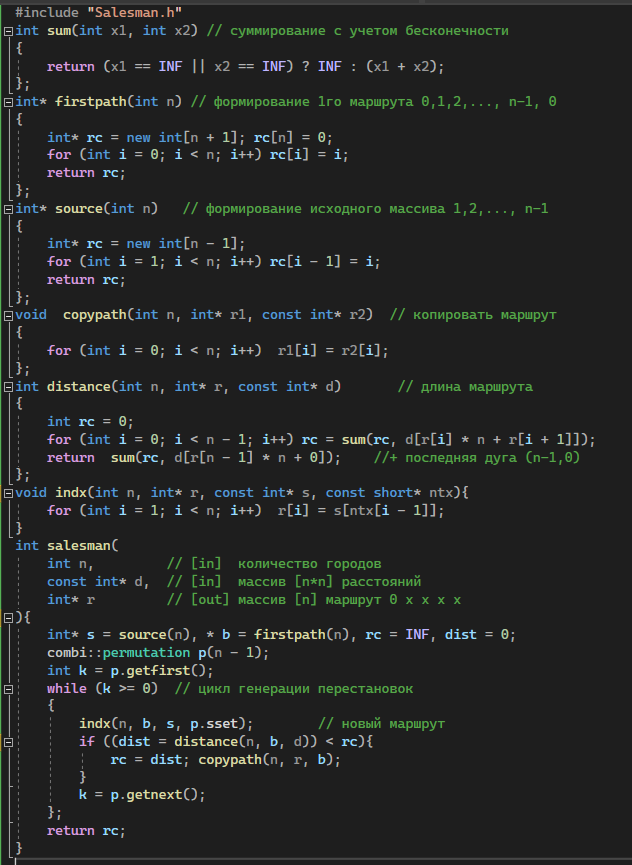


Рисунок 18 – Salesman.cpp

Результат выполнения программы представлен на рисунке 19.

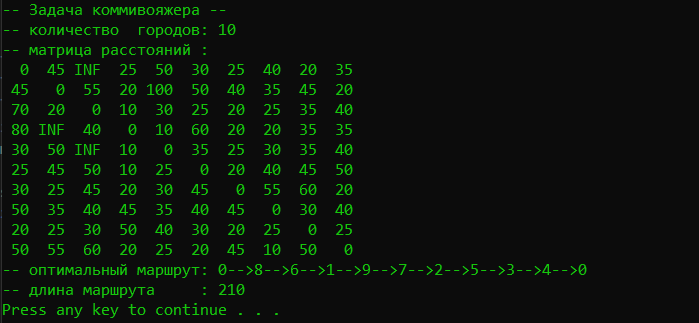


Рисунок 19 – результат работы программы

**Задание 6.** Исследовать зависимость времени вычисления необходимое для решения задачи от размерности задачи и результат в виде графика:

Результат работы программы представлен на рисунке 20

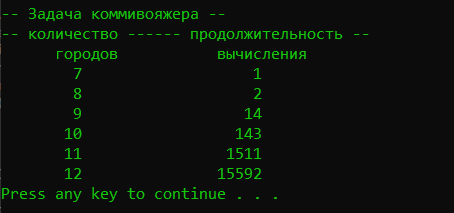


Рисунок 20 – Зависимость времени выполнения от количества городов

На рисунке 21 представлен график зависимости времени вычисления от количества городов.

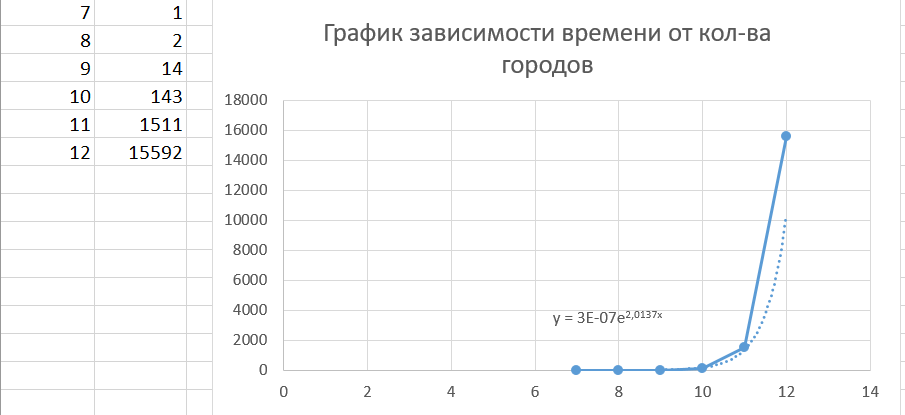


Рисунок 21 – график зависимости

**Вывод**: исходя из полученных данных и графика, можно заметить, что скорость выполнения программы экспоненциально возрастает при увеличении числа городов.

**Лабораторная работы №3**

**Метод ветвей и границ. Задача коммивояжера и методы её решения.**

**ЦЕЛЬ РАБОТЫ:** освоить общие принципы решения задач методом ветвей и границ, решить задачу о коммивояжере данным методом, сравнить полученное решение задачи с комбинаторным методом перестановок.

**Задание 1.** Сформулировать условие задачи коммивояжера с параметром.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Город | **1** | **2** | **3** | **4** | **5** |
| **1** | INF | 18 | 30 | INF | 9 |
| **2** | 9 | INF | 24 | 59 | 75 |
| **3** | 11 | 27 | INF | 86 | 58 |
| **4** | 26 | 49 | 36 | INF | 27 |
| **5** | 84 | 75 | 52 | 22 | INF |

Решение:

Для определения нижней границы множества воспользуемся **операцией редукции** или приведения матрицы по строкам, для чего необходимо в каждой строке матрицы D найти минимальный элемент.  
di = min(j) dij

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| i j | **1** | **2** | **3** | **4** | **5** | di |
| **1** | INF | 18 | 30 | INF | 9 | 9 |
| **2** | 9 | INF | 24 | 59 | 75 | 9 |
| **3** | 11 | 27 | INF | 86 | 58 | 11 |
| **4** | 26 | 49 | 36 | INF | 27 | 26 |
| **5** | 84 | 75 | 52 | 22 | INF | 22 |

Затем вычитаем di из элементов рассматриваемой строки.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| i j | **1** | **2** | **3** | **4** | **5** |
| **1** | INF | 9 | 21 | INF | 0 |
| **2** | 0 | INF | 15 | 50 | 66 |
| **3** | 0 | 16 | INF | 75 | 47 |
| **4** | 0 | 23 | 10 | INF | 1 |
| **5** | 62 | 53 | 30 | 0 | INF |

Такую же операцию редукции проводим по столбцам, для чего в каждом столбце находим минимальный элемент:  
dj = min(i) dij

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| i j | **1** | **2** | **3** | **4** | **5** |
| **1** | INF | 9 | 21 | INF | 0 |
| **2** | 0 | INF | 15 | 50 | 66 |
| **3** | 0 | 16 | INF | 75 | 47 |
| **4** | 0 | 23 | 10 | INF | 1 |
| **5** | 62 | 53 | 30 | 0 | INF |
| dj | 0 | 9 | 10 | 0 | 0 |

После вычитания минимальных элементов получаем полностью редуцированную матрицу, где величины di и dj называются **константами приведения**.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| i j | **1** | **2** | **3** | **4** | **5** |
| **1** | INF | 0 | 11 | INF | 0 |
| **2** | 0 | INF | 5 | 50 | 66 |
| **3** | 0 | 7 | INF | 75 | 47 |
| **4** | 0 | 14 | 0 | INF | 1 |
| **5** | 62 | 44 | 20 | 0 | INF |

Сумма констант приведения определяет нижнюю границу H:  
H = 9+9+11+26+22+0+9+10+0+0 = 96

**Определяем ребро ветвления** и разобьем все множество маршрутов относительно этого ребра на два подмножества (i,j) и (i\*,j\*).  
С этой целью для всех клеток матрицы с нулевыми элементами заменяем поочередно нули на INF и определяем для них сумму образовавшихся констант приведения, они приведены в скобках.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| i j | **1** | **2** | **3** | **4** | **5** | di |
| **1** | INF | 0(7) | 11 | INF | 0(1) | 0 |
| **2** | 0(5) | INF | 5 | 50 | 66 | 5 |
| **3** | 0(7) | 7 | INF | 75 | 47 | 7 |
| **4** | 0(0) | 14 | 0(5) | INF | 1 | 0 |
| **5** | 62 | 44 | 20 | 0(70) | INF | 20 |
| dj | 0 | 7 | 5 | 50 |  |  |

Наибольшая сумма констант приведения равна (20 + 50) = 70 для ребра (5,4), следовательно, множество разбивается на два подмножества (5,4) и (5\*,4\*).

**Исключение ребра** (5,4) проводим путем замены элемента d54 = 0 на INF, после чего осуществляем очередное приведение матрицы расстояний для образовавшегося подмножества (5\*,4\*), в результате получим редуцированную матрицу.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| i j | **1** | **2** | **3** | **4** | **5** | di |
| **1** | INF | 0 | 11 | INF | 0 | 0 |
| **2** | 0 | INF | 5 | 50 | 66 | 0 |
| **3** | 0 | 7 | INF | 75 | 47 | 0 |
| **4** | 0 | 14 | 0 | INF | 1 | 0 |
| **5** | 62 | 44 | 20 | INF | INF | 20 |
| dj | 0 | 0 | 0 | 50 | 0 | 70 |

Нижняя граница гамильтоновых циклов этого подмножества:  
H(5\*,4\*) = 96 + 70 = 166

**Включение ребра** (5,4) проводится путем исключения всех элементов 5-ой строки и 4-го столбца, в которой элемент d45 заменяем на М, для исключения образования негамильтонова цикла.  
В результате получим другую сокращенную матрицу (4 x 4), которая подлежит операции приведения.  
После операции приведения сокращенная матрица будет иметь вид:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| i j | **1** | **2** | **3** | **5** | di |
| **1** | INF | 0 | 11 | 0 | 0 |
| **2** | 0 | INF | 5 | 66 | 0 |
| **3** | 0 | 7 | INF | 47 | 0 |
| **4** | 0 | 14 | 0 | INF | 0 |
| dj | 0 | 0 | 0 | 0 | 0 |

Нижняя граница подмножества (5,4) равна:  
H(5,4) = 96 + 0 = 96 ≤ 166

Поскольку нижняя граница этого подмножества (5,4) меньше, чем подмножества (5\*,4\*), то ребро (5,4) включаем в маршрут с новой границей H = 96.

**Определяем ребро ветвления**.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| i j | **1** | **2** | **3** | **5** | di |
| **1** | INF | 0(7) | 11 | 0(47) | 0 |
| **2** | 0(5) | INF | 5 | 66 | 5 |
| **3** | 0(7) | 7 | INF | 47 | 7 |
| **4** | 0(0) | 14 | 0(5) | INF | 0 |
| dj | 0 | 7 | 5 | 47 |  |

max: d(1,5)=47

**Исключение ребра** (1,5): d15=INF.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| i j | **1** | **2** | **3** | **5** | di |
| **1** | INF | 0 | 11 | INF | 0 |
| **2** | 0 | INF | 5 | 66 | 0 |
| **3** | 0 | 7 | INF | 47 | 0 |
| **4** | 0 | 14 | 0 | INF | 0 |
| dj | 0 | 0 | 0 | 47 | 47 |

H(1\*,5\*) = 96 + 47 = 143

**Включение ребра** (1,5): d51=М.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| i j | **1** | **2** | **3** | di |
| **2** | 0 | INF | 5 | 0 |
| **3** | 0 | 7 | INF | 0 |
| **4** | 0 | 14 | 0 | 0 |
| dj | 0 | 7 | 0 | 7 |

H(1,5) = 96 + 7 = 103 ≤ 143

Ребро (1,5) включаем в маршрут с новой границей H=103.

**Определяем ребро ветвления**.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| i j | **1** | **2** | **3** | di |
| **2** | 0(5) | INF | 5 | 5 |
| **3** | 0(0) | 0(7) | INF | 0 |
| **4** | INF | 7 | 0(12) | 7 |
| dj | 0 | 7 | 5 |  |

max: d(4,3)=12.

**Исключение ребра** (4,3): d43=INF.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| i j | **1** | **2** | **3** | di |
| **2** | 0 | INF | 5 | 0 |
| **3** | 0 | 0 | INF | 0 |
| **4** | INF | 7 | INF | 7 |
| dj | 0 | 0 | 5 | 12 |

(4\*,3\*) = 103 + 12 = 115

**Включение ребра** (4,3): d34=INF.

|  |  |  |  |
| --- | --- | --- | --- |
| i j | **1** | **2** | di |
| **2** | 0 | INF | 0 |
| **3** | 0 | 0 | 0 |
| dj | 0 | 0 | 0 |

H(4,3) = 103 + 0 = 103 ≤ 115

Ребро (4,3) включаем в маршрут с новой границей H=103.  
В соответствии с этой матрицей включаем в гамильтонов маршрут ребра (2,1) и (3,2).  
В результате по дереву ветвлений гамильтонов цикл образуют ребра:  
(5,4), (4,3), (3,2), (2,1), (1,5),  
Длина маршрута = 103

Граф решения представлен на рисунке 3.1

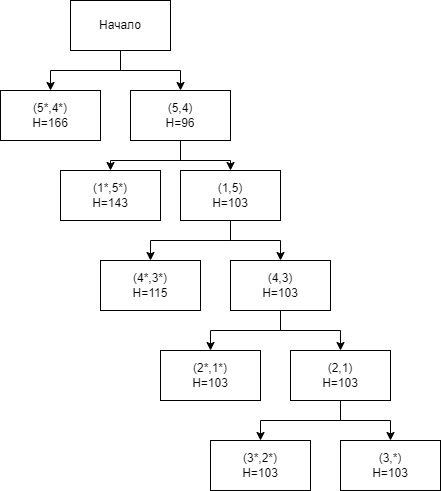


Рисунок 3.1 – граф решения

Результат решения задачи с помощью генератора перестановок представлен на рисунке 3.2

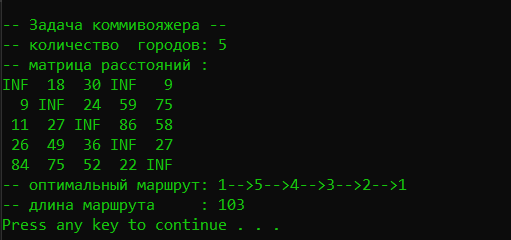


Рисунок 3.1 – результат решения с помощью генератора перестановок

**ВЫВОД**: были освоены общие принципы решения задач методом ветвей и границ, решена задача о коммивояжере данным методом, полученное решение было сравнено с комбинаторным методом перестановок.

**Лабораторная работа №4**

**Цель работы:** освоить общие принципы решения задач методом динамического программирования, сравнить полученные решения задач с рекурсивным методом.

**Практическая часть**

**Задание 1.**

На языке С++ сгенерировать случайным образом строку букв латинского алфавита *S1* длиной 300 символов и *S2*длиной 250. Листинг кода представлен на рисунке 4.1. Результат выполнения программы представлен на рисунке 4.2.

|  |
| --- |
| #define \_rand(min, max) ( rand() % ((max) - (min) + 1) + (min) )  void main()  {  srand(time(NULL));  char abc[25];  char s1[300], s2[250];  for (int i = 97, n = 0; i <= 122; ++i, ++n)  {  abc[n] = (char)i;  }  std::cout << "S1 = ";  for (int i = 0; i < 300; i++)  {  s1[i] = abc[\_rand(0, 25)];  if (i % 100 == 0)  std::cout << std::endl;  std::cout << s1[i];  }  std::cout << std::endl << std::endl << "S2 =";  for (int i = 0; i < 251; i++)  {  s2[i] = abc[\_rand(0, 25)];  if (i % 100 == 0)  std::cout << std::endl;  std::cout << s2[i];  }  std::cout << std::endl;  } |

Рисунок 4.1 – Генерация строк

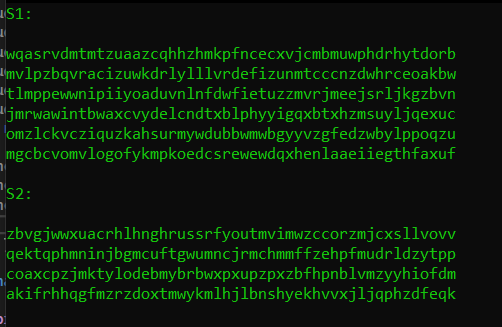


Рисунок 4.2 – Результат работы программы

**Задание 2.**

Вычислить двумя способами (рекурсивно и с помощью динамического программирования)  – Левенштейна для , где - длина строки ,  - строка, состоящая из первых  символов строки . (копии экрана и код вставить в отчет). Листинг кода представлен на рисунке 4.3. Результат выполнения программы представлен на рисунке 4.4.

|  |
| --- |
| int min3(int x1, int x2, int x3)  {  return std::min(std::min(x1, x2), x3);  }  int levenshtein(int lx, const char x[], int ly, const char y[])  {  int\*\* matr;  int w, left, top, left\_top;  matr = new int\* [lx];  for (int i = 0; i < lx; i++)  matr[i] = new int[ly];  matr[0][0] = 0;  for (int i = 1; i < lx; i++)  matr[i][0] = i;  for (int j = 1; j < ly; j++)  matr[0][j] = j;  for (int i = 1; i < lx; i++)  for (int j = 1; j < ly; j++)  {  w = x[i - 1] == y[j - 1] ? 0 : 1;  top = matr[i - 1][j];  left = matr[i][j - 1];  left\_top = matr[i - 1][j - 1];  matr[i][j] = std::min(left\_top + w, std::min(top + 1, left + 1));  }  return matr[lx - 1][ly - 1];  }  int levenshtein\_r(int lx, const char x[], int ly, const char y[])  {  int rc = 0;  if (lx == 0) rc = ly;  else if (ly == 0) rc = lx;  else if (lx == 1 && ly == 1 && x[0] == y[0]) rc = 0;  else if (lx == 1 && ly == 1 && x[0] != y[0]) rc = 1;  else rc = min3(  levenshtein\_r(lx - 1, x, ly, y) + 1,  levenshtein\_r(lx, x, ly - 1, y) + 1,  levenshtein\_r(lx - 1, x, ly - 1, y) + (x[lx - 1] == y[ly - 1] ? 0 : 1));  return rc;  }; |

Рисунок 4.3 – Вычисление расстояния Левенштейна

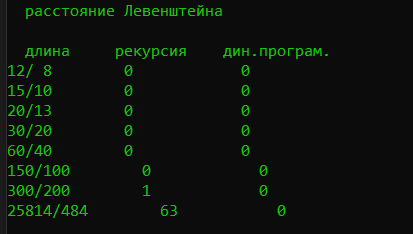


Рисунок 4.4 – Результат работы программы

**Задание 3.**

Выполнить сравнительный анализ времени затраченного на вычисление дистанции Левенштейна для двух методов решения. Построить графики зависимости времени вычисления от . На рисунке 4.5 представлены графики зависимости времени вычисления от *k*.

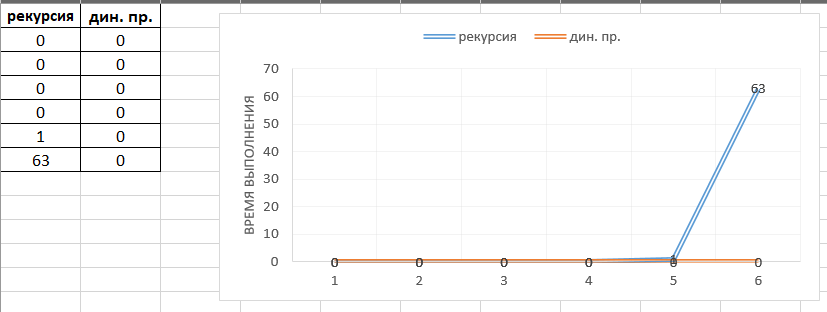


Рисунок 4.5 – Графики зависимости

**Задание 4.**

Реализовать вручную пример вычисления дистанции Левенштейна при помощи рекурсивного алгоритма (в соответствии с вариантом).

1.  
2.  
3.  
4.  
5.  

 = 5.

 = 4.

1.  

 = 4.

 = 3.

1.  
2.  
3.  
4.  

 = 3.

 = 2.

 = 2.

1.  

 = 3.

 = 2.

1.  

 =1.

1.  

 = 2.

 = 1.

 = 1.

1.  
2.  

 = 1.

 = 2.

 = 1.

1. 
2. 
3. 
4. 
5. 
6. 
7. 
8. 
9. 
10. 
11. 
12. /
13. 
14. 
15. 

**Задание 5.**

**Нечетные варианты**. Выполнить сравнительный анализ времени затраченного на решение задачи о наибольшей общей подпоследовательности для двух методов решения (рекурсивное решение, динамическое программирование). Две последовательности взять в соответствии с вариантом. Построить графики зависимости времени вычисления от .

**Отобразить ход решения в отчете**(по примеру из лекции) + код и копии экрана.

|  |
| --- |
| #include "stdafx.h"  #include <memory.h>  #include "MultyMatrix.h"  // расстановка скобок (рекурсия)  #define INFINITY 0x7fffffff  #define NINFINITY 0x80000000  int OptimalM(int i, int j, int n, const int c[], int\* s)  {  #define OPTIMALM\_S(x1,x2) (s[(x1-1)\*n+x2-1])  int o = INFINITY;  int bo = INFINITY;  if (i < j)  {  for (int k = i; k < j; k++)  {  bo = OptimalM(i, k, n, c, s) + OptimalM(k + 1, j, n, c, s) + c[i - 1] \* c[k] \* c[j];  if (bo < o)  {  o = bo;  OPTIMALM\_S(i, j) = k;  }  }  }  else o = 0;  return o;  #undef OPTIMALM\_S  };  // расстановка скобок (динамическое программирование)  int OptimalMD(int n, const int c[], int\* s)  {  #define OPTIMALM\_S(x1,x2) (s[(x1-1)\*n+x2-1])  #define OPTIMALM\_M(x1,x2) (M[(x1-1)\*n+x2-1])  int\* M = new int[n \* n], j = 0, q = 0;  for (int i = 1; i <= n; i++)  OPTIMALM\_M(i, i) = 0;  for (int l = 2; l <= n; l++)  {  for (int i = 1; i <= n - l + 1; i++)  {  j = i + l - 1;  OPTIMALM\_M(i, j) = INFINITY;  for (int k = i; k <= j - 1; k++)  {  q = OPTIMALM\_M(i, k) + OPTIMALM\_M(k + 1, j) + c[i - 1] \* c[k] \* c[j];  if (q < OPTIMALM\_M(i, j))  {  OPTIMALM\_M(i, j) = q;  OPTIMALM\_S(i, j) = k;  }  }  }  }  return OPTIMALM\_M(1, n);  #undef OPTIMALM\_M  #undef OPTIMALM\_S  }; |

Рисунок 4.6 – Решение задачи о расстановке скобок

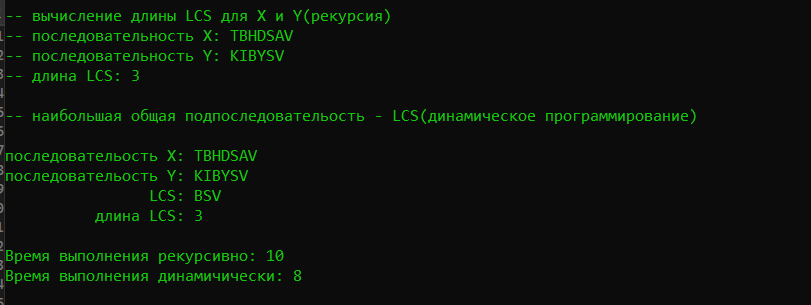


Рисунок 4.7 – Результат работы программы

На рисунке 4.8 представлен график зависимости от суммарной длинны двух подстрок и необходимом времени выполнения при помощи динамического алгоритма:

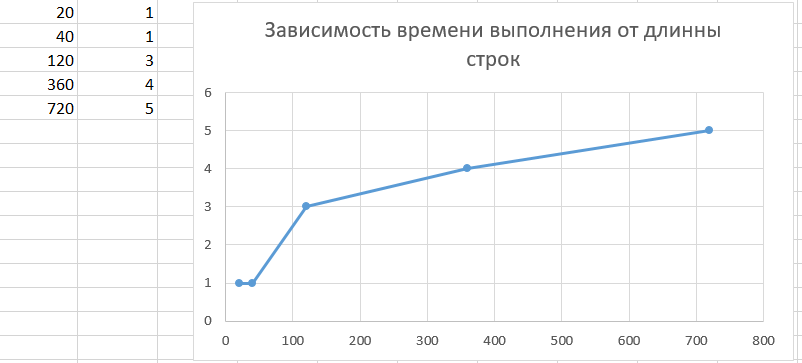
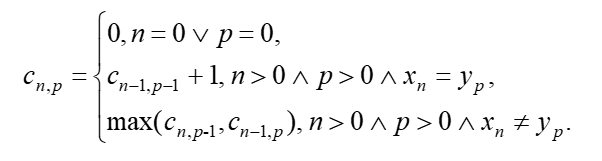


Рисунок 4.8 – График зависимости



**Алгоритм прохождения:**

1. Создаем двумерный массив размером (len(QVTWNHO) + 1) на (len(RQTWYK) + 1), где каждый элемент равен 0.

2. Проходим по каждой строке и столбцу массива, начиная с первого и до последнего.

3. Если значение строки или столбца равно 0, то элемент массива на пересечении этой строки и столбца остается равным 0.

4. Если символ строки и столбца равны, то элемент массива на пересечении этой строки и столбца равен значению элемента на предыдущей диагонали плюс 1, т.е. с\_(𝑛,𝑝) = c\_(𝑛−1,𝑝−1) + 1.

5. Если символ строки и столбца не равны, то элемент массива на пересечении этой строки и столбца равен максимуму между значением элемента на предыдущей строке и значением элемента на предыдущем столбце, т.е. c\_(𝑛,𝑝) = max⁡( с\_(𝑛,𝑝"−1" ), c\_(𝑛−1,𝑝)).

6. После прохода по всем элементам массива, наибольшая общая подпоследовательность будет равна значению элемента на пересечении последней строки и последнего столбца.

Алгоритм прохода будет выглядеть следующим образом:

1)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | T | B | H | D | S | A | V |
| K | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| I | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| B | 0 | 1 |  |  |  |  |  |
| Y |  |  |  |  |  |  |  |
| S |  |  |  |  |  |  |  |
| V |  |  |  |  |  |  |  |

2)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | T | B | H | D | S | A | V |
| K | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| I | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| B | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| Y | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| S | 0 | 1 | 1 | 1 | 2 |  |  |
| V | 0 |  |  |  |  |  |  |

3)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | T | B | H | D | S | A | V |
| K | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| I | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| B | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| Y | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| S | 0 | 1 | 1 | 1 | 2 | 2 | 2 |
| V | 0 | 1 | 1 | 1 | 2 | 2 | 3 |

KIBYSV

TBHDSAV

c6,7 = c5,6 + 1;

с5,7 = max(c5,6, c4,7);

с5,6 = max(c5,5, c4,6);

с5,5 = c4,4 + 1;

с4,7 = max(c4,6, c3,7);

с4,6 = max(c4,5, c3,6);

с4,5 = max(c4,4, c3,5);

с4,4 = max(c4,3, c3,4);

с4,3 = max(c4,2, c3,3);

с4,2 = max(c4,1, c3,2);

с4,1 = max(c4,0, c3,1) = c3,1;

c4,0 = 0;

с3,7 = max(c3,6, c2,7);

с3,6 = max(c3,5, c2,6);

с3,5 = max(c3,4, c2,5);

с3,4 = max(c3,3, c2,4);

с3,3 = max(c3,2, c2,3);

с3,2 = c2,1 + 1;

с3,1 = max(c3,0, c2,1) = c2,1;

с2,7 = max(c2,6, c1,7);

с2,6 = max(c2,5, c1,6);

с2,5 = max(c2,4, c1,5);

с2,4 = max(c2,3, c1,4);

с2,3 = max(c2,2, c1,3);

с2,2 = max(c2,1, c1,2);

с2,1 = max(c2,0, c1,1) = c1,1;

c2,0 = 0;

с1,7 = max(c1,6, c0,7) = c1,6;

c0,7 = 0;

с1,6 = max(c1,5, c0,6) = c1,5;

с1,5 = max(c1,4, c0,5) = c1,4;

с1,4 = max(c1,3, c0,4) = c1,3;

с1,3 = max(c1,2, c0,3) = c1,2;

с1,2 = max(c1,1, c0,2) = c1,1;

с1,1 = max(c1,0, c0,1) ;

с1,1 = max(0, 0) = 0;

с1,2 = max(0, 0) = 0;

с1,3 = max(0, 0) = 0;

с1,4 = max(0, 0) = 0;

с1,5 = max(0, 0) = 0;

с1,6 = max(0, 0) = 0;

с1,7 = max(0, 0) = 0;

с2,1 = max(0, 0) = 0;

с2,2 = max(0, 0) = 0;

с2,3 = max(0, 0) = 0;

с2,4 = max(0, 0) = 0;

с2,5 = max(0, 0) = 0;

с2,6 = max(0, 0) = 0;

с2,7 = max(0, 0) = 0;

с3,1 = max(0, 0) = 0;

с3,2 = 0 + 1 = 1;

с3,3 = max(1, 0) = 1;

с3,4 = max(1, 0) = 1;

с3,5 = max(1, 0) = 1;

с3,6 = max(1, 0) = 1;

с3,7 = max(1, 0) = 1;

с4,1 = max(0, 0) = 0;

с4,2 = max(0, 1) = 1;

с4,3 = max(1, 1) = 1;

с4,4 = max(1, 1) = 1;

с4,5 = max(1, 1) = 1;

с4,6 = max(1, 1) = 1;

с4,7 = max(1, 1) = 1;

с5,5 = 1 + 1 = 2;

с5,6 = max(2, 1) = 2;

с5,7 = max(2, 1) = 2;

с6,7 = 2 + 1 = 3;

**Вывод:** в результате выполнения лабораторной работы были освоены общие принципы решения задач методом динамического программирования. Были изучены его основные этапы и принципы работы алгоритмов. Были рассмотрены примеры решения задач методом динамического программирования и сравнены с рекурсивным методом.

**Лабораторная работа 5. ТРАНСПОРТНАЯ ЗАДАЧА**

**Цель работы:** Приобретение навыков решения открытой транспортной задачи

**Задание для выполнения:**

**Задание.** Решить транспортную задачу. Имеется 5 поставщиков продукции и 6 потребителей. Величина запасов, потребностей и стоимость затрат на перевозку продукции взять в соответствии с вариантом (*N*). Оформить отчет.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ПОТРЕБИТЕЛИ  ПОСТАВЩИКИ | 1 | 2 | 3 | 4 | 5 | 6 | ЗАПАСЫ |
| 1 | **21** | **11** | **15** | **12** | **20** | **10** | **177** |
| 2 | **19** | **9** | **17** | **14** | **16** | **22** | **122** |
| 3 | **10** | **14** | **20** | **17** | **11** | **20** | **159** |
| 4 | **13** | **19** | **19** | **12** | **22** | **11** | **168** |
| 5 | **12** | **20** | **18** | **9** | **19** | **13** | **109** |
| ПОТРЕБНОСТИ | **152** | **116** | **140** | **202** | **104** | **172** |  |

Запасы: 177+122+159+168+109=735;

Потребности: 152+116+140+202+104+172=886

Задача открытая, добавляем поставщика (886 – 737 = 151)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ПОТРЕБИТЕЛИ  ПОСТАВЩИКИ | 1 | 2 | 3 | 4 | 5 | 6 | ЗАПАСЫ |
| 1 | **21** | **11** | **15** | **12** | **20** | **10** | **177** |
| 2 | **19** | **9** | **17** | **14** | **16** | **22** | **122** |
| 3 | **10** | **14** | **20** | **17** | **11** | **20** | **159** |
| 4 | **13** | **19** | **19** | **12** | **22** | **11** | **168** |
| 5 | **12** | **20** | **18** | **9** | **19** | **13** | **109** |
| 6 | **0** | **0** | **0** | **0** | **0** | **0** | **151** |
| ПОТРЕБНОСТИ | **152** | **116** | **140** | **202** | **104** | **172** |  |

Составление опорного плана

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ПОТРЕБИТЕЛИ  ПОСТАВЩИКИ | 1 | 2 | 3 | 4 | 5 | 6 | ЗАПАСЫ |
| 1 | **21** | **11** | **15** | **12** | **20** | **10** | **177** |
| 2 | **19** | **9** | **17** | **14** | **16** | **22** | **122** |
| 3 | **10** | **14** | **20** | **17** | **11** | **20** | **159** |
| 4 | **13** | **19** | **19** | **12** | **22** | **11** | **168** |
| 5 | **12** | **20** | **18** | **9** | **19** | **13** | **109** |
| 6 | **0** | **0** | **0** | **0** | **0** | **0** | **151** |
| ПОТРЕБНОСТИ | **152** | **116** | **140** | **202** | **104** | **172** |  |

x22 = min(116, 122) = 116

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ПОТРЕБИТЕЛИ  ПОСТАВЩИКИ | 1 | 2 | 3 | 4 | 5 | 6 | ЗАПАСЫ |
| 1 | **21** | **11** | **15** | **12** | **20** | **10** | **177** |
| 2 | **19** | **9|116** | **17** | **14** | **16** | **22** | **6** |
| 3 | **10** | **14** | **20** | **17** | **11** | **20** | **159** |
| 4 | **13** | **19** | **19** | **12** | **22** | **11** | **168** |
| 5 | **12** | **20** | **18** | **9** | **19** | **13** | **109** |
| 6 | **0** | **0** | **0** | **0** | **0** | **0** | **151** |
| ПОТРЕБНОСТИ | **152** | **0** | **140** | **202** | **104** | **172** |  |

x54 = min(202, 109) = 109;

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ПОТРЕБИТЕЛИ  ПОСТАВЩИКИ | 1 | 2 | 3 | 4 | 5 | 6 | ЗАПАСЫ |
| 1 | **21** | **11** | **15** | **12** | **20** | **10** | **177** |
| 2 | **19** | **9|116** | **17** | **14** | **16** | **22** | **6** |
| 3 | **10|152** | **14** | **20** | **17** | **11** | **20** | **159** |
| 4 | **13** | **19** | **19** | **12** | **22** | **11** | **168** |
| 5 | **12** | **20** | **18** | **9|109** | **19** | **13** | **0** |
| 6 | **0** | **0** | **0** | **0** | **0** | **0** | **151** |
| ПОТРЕБНОСТИ | **152** | **0** | **140** | **93** | **104** | **172** |  |

x31 = min(159, 152) = 152;

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ПОТРЕБИТЕЛИ  ПОСТАВЩИКИ | 1 | 2 | 3 | 4 | 5 | 6 | ЗАПАСЫ |
| 1 | **21** | **11** | **15** | **12** | **20** | **10** | **177** |
| 2 | **19** | **9|116** | **17** | **14** | **16** | **22** | **6** |
| 3 | **10|152** | **14** | **20** | **17** | **11** | **20** | **7** |
| 4 | **13** | **19** | **19** | **12** | **22** | **11** | **168** |
| 5 | **12** | **20** | **18** | **9|109** | **19** | **13** | **0** |
| 6 | **0** | **0** | **0** | **0** | **0** | **0** | **151** |
| ПОТРЕБНОСТИ | **0** | **0** | **140** | **93** | **104** | **172** |  |

x16 = min(172, 177) = 172;

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ПОТРЕБИТЕЛИ  ПОСТАВЩИКИ | 1 | 2 | 3 | 4 | 5 | 6 | ЗАПАСЫ |
| 1 | **21** | **11** | **15** | **12** | **20** | **10|172** | **5** |
| 2 | **19** | **9|116** | **17** | **14** | **16** | **22** | **6** |
| 3 | **10|152** | **14** | **20** | **17** | **11** | **20** | **7** |
| 4 | **13** | **19** | **19** | **12** | **22** | **11** | **168** |
| 5 | **12** | **20** | **18** | **9|109** | **19** | **13** | **0** |
| 6 | **0** | **0** | **0** | **0** | **0** | **0** | **151** |
| ПОТРЕБНОСТИ | **0** | **0** | **140** | **93** | **104** | **0** |  |

x35 = min(104, 7) = 7;

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ПОТРЕБИТЕЛИ  ПОСТАВЩИКИ | 1 | 2 | 3 | 4 | 5 | 6 | ЗАПАСЫ |
| 1 | **21** | **11** | **15** | **12** | **20** | **10|172** | **5** |
| 2 | **19** | **9|116** | **17** | **14** | **16** | **22** | **6** |
| 3 | **10|152** | **14** | **20** | **17** | **11|7** | **20** | **0** |
| 4 | **13** | **19** | **19** | **12** | **22** | **11** | **168** |
| 5 | **12** | **20** | **18** | **9|109** | **19** | **13** | **0** |
| 6 | **0** | **0** | **0** | **0** | **0** | **0** | **151** |
| ПОТРЕБНОСТИ | **0** | **0** | **140** | **93** | **97** | **0** |  |

x14 = min(93, 5) = 5;

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ПОТРЕБИТЕЛИ  ПОСТАВЩИКИ | 1 | 2 | 3 | 4 | 5 | 6 | ЗАПАСЫ |
| 1 | **21** | **11** | **15** | **12|5** | **20** | **10|172** | **0** |
| 2 | **19** | **9|116** | **17** | **14** | **16** | **22** | **6** |
| 3 | **10|152** | **14** | **20** | **17** | **11|7** | **20** | **0** |
| 4 | **13** | **19** | **19** | **12** | **22** | **11** | **168** |
| 5 | **12** | **20** | **18** | **9|109** | **19** | **13** | **0** |
| 6 | **0** | **0** | **0** | **0** | **0** | **0** | **151** |
| ПОТРЕБНОСТИ | **0** | **0** | **140** | **88** | **97** | **0** |  |

x14 = min(93, 5) = 5;

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ПОТРЕБИТЕЛИ  ПОСТАВЩИКИ | 1 | 2 | 3 | 4 | 5 | 6 | ЗАПАСЫ |
| 1 | **21** | **11** | **15** | **12|5** | **20** | **10|172** | **0** |
| 2 | **19** | **9|116** | **17** | **14** | **16** | **22** | **6** |
| 3 | **10|152** | **14** | **20** | **17** | **11|7** | **20** | **0** |
| 4 | **13** | **19** | **19** | **12** | **22** | **11** | **168** |
| 5 | **12** | **20** | **18** | **9|109** | **19** | **13** | **0** |
| 6 | **0** | **0** | **0** | **0** | **0** | **0** | **151** |
| ПОТРЕБНОСТИ | **0** | **0** | **140** | **88** | **97** | **0** |  |

x44 = min(88, 168) = 88;

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ПОТРЕБИТЕЛИ  ПОСТАВЩИКИ | 1 | 2 | 3 | 4 | 5 | 6 | ЗАПАСЫ |
| 1 | **21** | **11** | **15** | **12|5** | **20** | **10|172** | **0** |
| 2 | **19** | **9|116** | **17** | **14** | **16** | **22** | **6** |
| 3 | **10|152** | **14** | **20** | **17** | **11|7** | **20** | **0** |
| 4 | **13** | **19** | **19** | **12** | **22** | **11** | **168** |
| 5 | **12** | **20** | **18** | **9|109** | **19** | **13** | **0** |
| 6 | **0** | **0** | **0** | **0** | **0** | **0** | **151** |
| ПОТРЕБНОСТИ | **0** | **0** | **140** | **88** | **97** | **0** |  |

x44 = min(88, 168) = 88;

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ПОТРЕБИТЕЛИ  ПОСТАВЩИКИ | 1 | 2 | 3 | 4 | 5 | 6 | ЗАПАСЫ |
| 1 | **21** | **11** | **15** | **12|5** | **20** | **10|172** | **0** |
| 2 | **19** | **9|116** | **17** | **14** | **16** | **22** | **6** |
| 3 | **10|152** | **14** | **20** | **17** | **11|7** | **20** | **0** |
| 4 | **13** | **19** | **19** | **12|88** | **22** | **11** | **80** |
| 5 | **12** | **20** | **18** | **9|109** | **19** | **13** | **0** |
| 6 | **0** | **0** | **0** | **0** | **0** | **0** | **151** |
| ПОТРЕБНОСТИ | **0** | **0** | **140** | **0** | **97** | **0** |  |

x25 = min(97, 6) = 6;

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ПОТРЕБИТЕЛИ  ПОСТАВЩИКИ | 1 | 2 | 3 | 4 | 5 | 6 | ЗАПАСЫ |
| 1 | **21** | **11** | **15** | **12|5** | **20** | **10|172** | **0** |
| 2 | **19** | **9|116** | **17** | **14** | **16|6** | **22** | **0** |
| 3 | **10|152** | **14** | **20** | **17** | **11|7** | **20** | **0** |
| 4 | **13** | **19** | **19** | **12|88** | **22** | **11** | **80** |
| 5 | **12** | **20** | **18** | **9|109** | **19** | **13** | **0** |
| 6 | **0** | **0** | **0** | **0** | **0** | **0** | **151** |
| ПОТРЕБНОСТИ | **0** | **0** | **140** | **0** | **91** | **0** |  |

x43 = min(140, 80) = 80;

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ПОТРЕБИТЕЛИ  ПОСТАВЩИКИ | 1 | 2 | 3 | 4 | 5 | 6 | ЗАПАСЫ |
| 1 | **21** | **11** | **15** | **12|5** | **20** | **10|172** | **0** |
| 2 | **19** | **9|116** | **17** | **14** | **16|6** | **22** | **0** |
| 3 | **10|152** | **14** | **20** | **17** | **11|7** | **20** | **0** |
| 4 | **13** | **19** | **19|80** | **12|88** | **22** | **11** | **0** |
| 5 | **12** | **20** | **18** | **9|109** | **19** | **13** | **0** |
| 6 | **0** | **0** | **0** | **0** | **0** | **0** | **151** |
| ПОТРЕБНОСТИ | **0** | **0** | **60** | **0** | **91** | **0** |  |

x63 = min(60, 151) = 60;

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ПОТРЕБИТЕЛИ  ПОСТАВЩИКИ | 1 | 2 | 3 | 4 | 5 | 6 | ЗАПАСЫ |
| 1 | **21** | **11** | **15** | **12|5** | **20** | **10|172** | **0** |
| 2 | **19** | **9|116** | **17** | **14** | **16|6** | **22** | **0** |
| 3 | **10|152** | **14** | **20** | **17** | **11|7** | **20** | **0** |
| 4 | **13** | **19** | **19|80** | **12|88** | **22** | **11** | **0** |
| 5 | **12** | **20** | **18** | **9|109** | **19** | **13** | **0** |
| 6 | **0** | **0** | **0|60** | **0** | **0** | **0** | **91** |
| ПОТРЕБНОСТИ | **0** | **0** | **0** | **0** | **91** | **0** |  |

x65 = min(91, 91) = 91;

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ПОТРЕБИТЕЛИ  ПОСТАВЩИКИ | 1 | 2 | 3 | 4 | 5 | 6 | ЗАПАСЫ |
| 1 | **21** | **11** | **15** | **12|5** | **20** | **10|172** | **0** |
| 2 | **19** | **9|116** | **17** | **14** | **16|6** | **22** | **0** |
| 3 | **10|152** | **14** | **20** | **17** | **11|7** | **20** | **0** |
| 4 | **13** | **19** | **19|80** | **12|88** | **22** | **11** | **0** |
| 5 | **12** | **20** | **18** | **9|109** | **19** | **13** | **0** |
| 6 | **0** | **0** | **0|60** | **0** | **0|91** | **0** | **0** |
| ПОТРЕБНОСТИ | **0** | **0** | **0** | **0** | **0** | **0** |  |

есть m + n – 1 = 11 базисных переменных

Первое допустимое решение

x14 = 5, x16 = 172, x22 = 116, x25 = 6, x31 = 152, x35 = 7, x43 = 80, x44 = 88, x54 = 9,

x63 = 60, x65 = 91;

Значение функции цели

Z = 5\*12 + 172\*10 + 116\*9 + 6\*16 + 152\*10 + 7\*11 + 80\*19 + 88\*12 + 109\*9 + 60\*0 + 91\*0 = 8074

Метод потенциалов

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ПОТРЕБИТЕЛИ  ПОСТАВЩИКИ | 1 | 2 | 3 | 4 | 5 | 6 | ЗАПАСЫ | V |
| 1 | **21** | **11** | **15** | **12|5** | **20** | **10|172** | **0** | **U1=0** |
| 2 | **19** | **9|116** | **17** | **14** | **16|6** | **22** | **0** | **U2=-3** |
| 3 | **10|152** | **14** | **20** | **17** | **11|7** | **20** | **0** | **U3=-8** |
| 4 | **13** | **19** | **19|80** | **12|88** | **22** | **11** | **0** | **U4=0** |
| 5 | **12** | **20** | **18** | **9|109** | **19** | **13** | **0** | **U5=-3** |
| 6 | **0** | **0** | **0|60** | **0** | **0|91** | **0** | **0** | **U6=-19** |
| ПОТРЕБНОСТИ | **0** | **0** | **0** | **0** | **0** | **0** |  |  |
| U | **v1=18** | **v2=12** | **v3=19** | **v4=12** | **v5=19** | **v6=10** |  |  |

Потенциалы

u1 + v6 = 10; 0 + v6 = 10; v6 = 10;

u1 + v4 = 12; 0 + v4 = 12; v4 = 12;

u4 + v4 = 12; u4 + 12 = 12; u4 = 0;

u5 + v4 = 9; u5 + 12 = 9 u5 = -3;

u4 + v3 = 19; 0 + v3 = 19; v3 = 19;

u6 + v3 = 0; 19 + v3 = 0; u6 = -19;

u6 + v5 = 0; -19 + v5 = 0; v5 = 19;

u3 + v5 = 11; u3 + 19 = 11; u3 = -8;

u2 + v5 = 16; u3 + 19 = 16; u2 = -3;

u2 + v2 = 9; -3 + v2 = 9; v2 = 12;

u3 + v1 = 10; -8 + v1 = 10; v1 = 18;

Для свободных клеток

x11 = u1 + v1 – c11 = 0 + 18 – 21 = -3

x12 = u1 + v2 – c12 = 0 + 12 – 11 = 1

x13 = u1 + v3 – c13 = 0 + 19 – 15 = 4

x15 = u1 + v5 – c15 = 0 + 19 – 20 = -1

x21 = u2 + v1 – c21 = -3 + 18 – 19 = -4

x23 = -3 + 19 – 17 = -5

x24 = -3 + 12 – 14 = -5

x26 = -3 + 10 – 22 = - 15

x32 = -8 + 12 – 14 = -10

x33 = -8 + 19 – 20 = -9

x34 = -8 + 12 – 17 = -13

x36 = -8 + 10 – 20 = -18

x41 = 0 + 18 – 13 = 5

x42 = 0 + 12 – 19 = -5

x45 = 0 + 19 – 22 = -3

x46 = 0 + 10 – 11 = -1

x51 = -3 + 18 – 12 = 3

x52 = -3 + 12 – 20 = -11

x53 = -3 + 19 – 18 = -2

x55 = -3 + 19 – 19 = -3

x56 = -3 + 10 – 13 = 0

x61 = -19 + 18 – 0 = -1

x62 = -19 + 12 – 0 = -7

x64 = -19 + 12 – 0 = -7

x66 = -19 + 10 – 0 = -9

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ПОТРЕБИТЕЛИ  ПОСТАВЩИКИ | 1 | 2 | 3 | 4 | 5 | 6 | ЗАПАСЫ | u |
| 1 | **21** | **11** | **15** | **12|5** | **20** | **10|172** | **0** | **u1= 0** |
| 2 | **19** | **9|116** | **17** | **14** | **16|6** | **22** | **0** | **u2=-3** |
| 3 | **10|152(-** | **14** | **20** | **17** | **11|7(+** | **20** | **0** | **u3=-8** |
| 4 | **13(+** | **19** | **19|80(-** | **12|88** | **22** | **11** | **0** | **u4=0** |
| 5 | **12** | **20** | **18** | **9|109** | **19** | **13** | **0** | **u5=-3** |
| 6 | **0** | **0** | **0|60(+** | **0** | **0|91(-** | **0** | **0** | **u6=-19** |
| ПОТРЕБНОСТИ | **0** | **0** | **0** | **0** | **0** | **0** |  |  |
| v | **v1 =18** | **v2 =12** | **v3 =19** | **v4 =12** | **v5 =19** | **v6 =10** |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ПОТРЕБИТЕЛИ  ПОСТАВЩИКИ | 1 | 2 | 3 | 4 | 5 | 6 | ЗАПАСЫ | u |
| 1 | **21** | **11** | **15** | **12|5** | **20** | **10|172** | **0** | **u1= 0** |
| 2 | **19** | **9|116** | **17** | **14** | **16|6** | **22** | **0** | **u2=2** |
| 3 | **10|70** | **14** | **20** | **17** | **11|87** | **20** | **0** | **u3=-3** |
| 4 | **13|80** | **19** | **19|0** | **12|88** | **22** | **11** | **0** | **u4=0** |
| 5 | **12** | **20** | **18** | **9|109** | **19** | **13** | **0** | **u5=-3** |
| 6 | **0** | **0** | **0|140** | **0** | **0|11** | **0** | **0** | **u6=-14** |
| ПОТРЕБНОСТИ | **0** | **0** | **0** | **0** | **0** | **0** |  |  |
| v | **v1 =13** | **v2 =7** | **v3 =14** | **v4 =12** | **v5 =14** | **v6 =10** |  |  |

Потенциалы

u1 + v6 = 10; 0 + v6 = 10; v6 = 10;

u1 + v4 = 12; 0 + v4 = 12; v4 = 12;

u4 + v4 = 12; u4 + 12 = 12; u4 = 0;

u5 + v4 = 9; u5 + 12 = 9; u5 = -3;

u4 + v1 = 13; 0 + v1 = 13; v1 = 13;

u3 + v1 = 10; u3 + 13 = 10; u3 = -3;

u3 + v5 = 11; -3 + v5 = 11; v5 = 14;

u2 + v5 = 16; u2 + 14 = 16; u2 = 2;

u2 + v2 = 9; 2 + v2 = 9; v2 = 7;

u6 + v5 = 0; u6 + 14 = 0; u6 = -14;

u6 + v3 = 0; -14 + v3 = 0; v3 = 14;

Для свободных клеток

x11 = 0 + 13 – 21 = -8

x12 = 0 + 7– 11 = -3

x13 = 0 + 14 – 15 = -1

x15 = 0 + 14 – 20 = -6

x21 = 2 + 12 – 19 = -5

x23 = 2 + 14 – 17 = -1

x24 = 2 + 12 – 14 = 0

x26 = 2 + 10 – 22 = -10

x32 = - 3 + 7 – 14 = -10

x33 = -3 + 14 – 20 = -9

x34 = -3 + 12 – 17 = -8

x36 = -3 + 10 – 20 = -7

x42 = 0 + 7 – 19 = -8

x43 = 0 + 14 – 19 = -5

x45 = 0 + 14 – 22 = -8

x46 = 0 + 10 – 11 = -1

x51 = -3 + 13 – 12 = -2

x52 = -3 + 7 – 20 = -10

x53 = -3 + 14 – 18 = -7

x55 = -3 + 14 – 19 = -8

x56 = -3 + 10 – 13 = 0

x61 = -14 + 13 – 0 = -1

x62 = -14 + 7 – 0 = -7

x64 = -14 + 12 – 0 = -2

x66 = -14 + 10 – 0 = -4

Опорный план является оптимальным, так все оценки свободных клеток удовлетворяют условию ui + vj ≤ cij.

Минимальные затраты составят: F(x) = 12\*5 + 10\*172 + 9\*116 + 6\*16 + 10\*70 + 11\*87 + 13\*80 + 12\*88 + 9\*109 + 0\*140 + 0\*11 = 7654

**Вывод:** В ходе выполнения лабораторной работы были приобретены навыки решения открытой транспортной задачи. Для решения задачи был применен метод наименьшей стоимости и метод потенциалов, которые позволяют определить оптимальный план перевозок при минимальных затратах. Были проведены расчеты с использованием данных методов, что позволило получить оптимальный план перевозок.