**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**ФЕДЕРАЛЬНОЕ ГОСУДРАСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ   
ТЕХНОЛОГИЧСЕКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»  
(БГТУ им В.Г.Шухова)**

Кафедра программного обеспечения вычислительной техники и   
автоматизированных систем

Лабораторная работа №2.2  
дисциплина: **Дискретная математика**  
тема: «**Задачи выбора**»

Выполнил: ст. группы ВТ-11   
.  
Проверил: Рязанов Ю.Д.

Белгород 2019

**Цель работы:** приобрести практические навыки в использовании алгоритмов порождения комбинаторных объектов при проектировании алгоритмов решения задача выбора.

1. Ознакомиться с задачей

Задача: Каждая из n деталей последовательно обрабатывается на m

станках (сначала на 1-м, затем на 2-м и т.д.). Для каждой детали известно время обработки на каждом станке. Определить, в какой последовательности необходимо обрабатывать детали, чтобы время обработки всех деталей было минимальным.

1. Определить класс комбинаторных объектов, содержащих решение задачи (траекторию задачи).

Класс используемых комбинаторных объектов в задаче – перестановки.

1. Определить, что в задаче является функционалом и способ его вычисления.

Функционал данной задачи – время обработки всех деталей.

1. Определить способ распознавания решения по значению функционала.

Значение функционала будет нахождение время выполнения меньшее, чем то, которое есть – это будет являться новым решением. И так пока не будет найдено самое минимальное.

1. Реализовать алгоритм решения задачи.

#include <stdio.h>

#include <stdlib.h>

#include <locale.h>

#define M 10

#define N 10

#define Z 5040

size\_t m, n;

int s;

void set\_init(int w[])

{

for (int i = 0; i < n; i++)

w[i] = i + 1;

}

void output\_set (int p[])

{

int m = 0;

for (int i = 0; i < n; i++)

printf("%d ", p[i]);

if (n == 0)

printf("Пустое множество");

puts("");

}

int resid(int m[], int n, int b[], int x)

{

int len\_b = 0;

for (int i = 0; i < n; i++)

if (m[i] != x){

b[len\_b] = m[i];

++len\_b;

}

return len\_b;

}

void zeroing\_arr (int k[])

{

for (int i = 0; i < M; i++)

k[i] = 0;

}

int result\_time (int k[])

{

int max\_time = k[0];

for (int i = 1; i < m; i++){

if (k[i] > max\_time)

max\_time = k[i];

}

return max\_time;

}

int counting\_time (int a[][N], int p[])

{

int k[M];

zeroing\_arr(k);

for (int j = 0; j < n; j++){

for (int i = 0; i < m-1; i++){

k[i] += a[i][p[j]-1];

for (int l = i+1; l < m; l++, i++)

if (k[l] < k[i]){

k[l] = k[i] + a[l][p[j]-1];

}else{

k[l] += a[l][p[j]-1];

}

}

}

return result\_time(k);

}

int min\_time\_work (int z[], int \*min\_pos)

{

int min\_time = z[0];

\*min\_pos = 0;

for (int i = 1; i < s; i++){

if (z[i] < min\_time){

min\_time = z[i];

\*min\_pos = i;

}

}

return min\_time;

}

void permutation(int p[], int i, int w[], int t, int a[][N], int z[], int r [][N])

{

for (int j = 0; j < t; j++){

p[i] = w[j];

if (i == n-1){

for (int l = 0; l < n; l++)

r[s][l] = p[l];

z[s] = counting\_time(a, p);

++s;

}else{

int b[t-1], len\_b;

len\_b = resid(w, t, b, p[i]);

permutation(p, i+1, b, len\_b, a, z, r);

}

}

}

void input\_characteristics(int a[][N])

{

printf ("Введите время обработки\n");

for (int i = 0; i < m; i++){

for (int j = 0; j < n; j++){

printf ("На %d станке %d детали:\n", i+1, j+1);

scanf ("%d", &a[i][j]);

}

}

}

int main()

{

setlocale(LC\_ALL, "rus");

printf ("Введите количество станков:\n");

scanf ("%d", &m);

printf ("Введите количество деталей:\n");

scanf ("%d", &n);

int a[M][N];

input\_characteristics (a);

int w[N];

int z[Z];

set\_init(w);

int p[N];

int i = 0;

int t = n;

s = 0;

int r[Z][N];

int min\_pos;

permutation(p, i, w, t, a, z, r);

printf ("Самое быстрое выполнение работы: %d\n", min\_time\_work (z, &min\_pos));

printf ("Для последовательности: ");

output\_set(r[min\_pos]);

printf("Количество перестановок: %d\n", s);

}

1. Подготовить тестовые данные и решить задачу

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Номер детали | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Время обработки на станке 1 | 20 | 16 | 8 | 6 | 14 | 14 | 20 |
| Время обработки на станке 2 | 21 | 18 | 12 | 10 | 10 | 8 | 18 |



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Номер детали | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Время обработки на станке 1 | 17 | 32 | 7 | 10 | 36 | 24 | 25 |
| Время обработки на станке 2 | 21 | 17 | 15 | 15 | 23 | 6 | 19 |
| Время обработки на станке 3 | 3 | 12 | 26 | 32 | 31 | 5 | 6 |



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Номер детали | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Время обработки на станке 1 | 20 | 16 | 8 | 6 | 14 | 14 | 20 |
| Время обработки на станке 2 | 21 | 18 | 12 | 10 | 10 | 8 | 18 |
| Время обработки на станке 3 | 14 | 23 | 26 | 24 | 20 | 17 | 4 |
| Время обработки на станке 4 | 16 | 35 | 7 | 7 | 4 | 15 | 12 |

