Міністерство освіти та науки України

Вінницький національний технічний університет

Факультет інформаційних технологій та комп’ютерної інженерії

Кафедра комп’ютерних наук

Лабораторна робота №1

З дисципліни: «Дискретна математика»

Тема: «Розробка алгоритму і програми для розв’язання задачі про покриття на множинах методом повного та граничного перебору»

Виконали студенти:

групи 2КН 18-б

І курсу

Воронков О.І.

Максимменюк М.Р.

Перевірив викладач:

Ваховська Л. М.

Вінниця-2019

**Мета:** набути навиків застосування методу повного перебору та методу граничного перебору для знаходження найкоротшого і мінімального покриттів.

**Завдання:** вирішити задачу про покриття методом повного перебору та методом граничного перебору згідно свого варіанту.

**Варіанти 8, 11**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **8** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | | **A** | **0** | **1** | **0** | **0** | **0** | **1** | **0** | **1** | **0** | | **B** | **0** | **1** | **0** | **1** | **0** | **0** | **1** | **0** | **1** | | **C** | **0** | **0** | **0** | **1** | **1** | **0** | **0** | **1** | **1** | | **D** | **0** | **0** | **1** | **0** | **0** | **1** | **0** | **0** | **1** | | **E** | **1** | **0** | **0** | **0** | **1** | **0** | **1** | **0** | **0** | | **F** | **1** | **0** | **1** | **1** | **0** | **0** | **0** | **0** | **0** | | **G** | **1** | **0** | **1** | **0** | **1** | **1** | **0** | **1** | **0** |   A=1.B=3.C=2.D=2.E=1.F=2.G=3 | |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **11** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | | **A** | **0** | **0** | **1** | **0** | **1** | **0** | **0** | **0** | **1** | | **B** | **0** | **1** | **1** | **0** | **0** | **1** | **1** | **0** | **0** | | **C** | **0** | **1** | **1** | **1** | **0** | **0** | **0** | **0** | **1** | | **D** | **1** | **0** | **0** | **1** | **1** | **0** | **0** | **1** | **1** | | **E** | **0** | **0** | **0** | **0** | **1** | **1** | **0** | **1** | **0** | | **F** | **1** | **1** | **0** | **1** | **0** | **1** | **1** | **0** | **0** | | **G** | **1** | **0** | **0** | **0** | **0** | **0** | **1** | **1** | **0** |   A=1 , B = 3, C = 2, D=3,E=1,F=2, G=2 |

Змодулюємо блок-схему роботи програми для розв’язання задачі на покриття

**Покриття чи ні**

**Виводимо елементи без покриття**

**Виводимо покриття**

**Виведення даних**

**Підключення бібліотек мови С/С++**

**Введення даних**

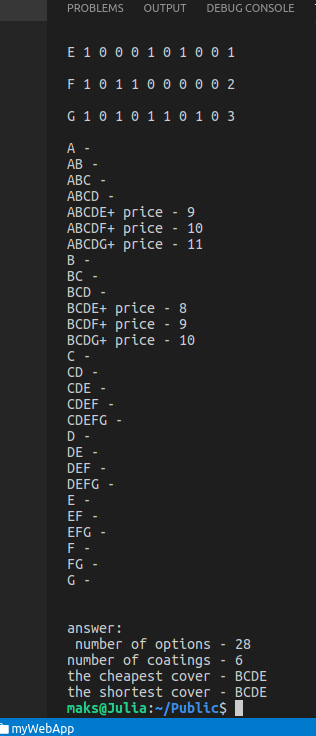
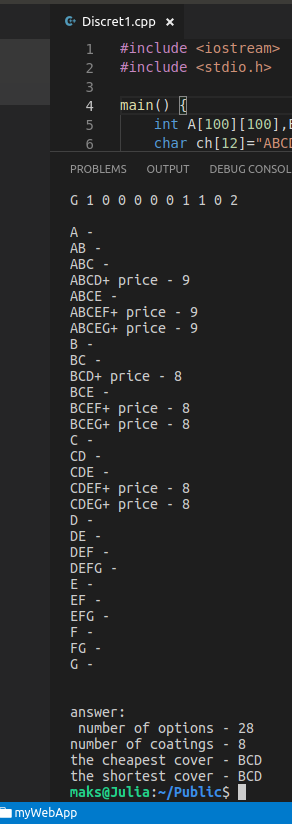
**Виведимо введен данні**

**Зчитуємо рядки з числами**

Рисунок 1 -- Схема програми побудови покриттів на множинах методом граничного перебору

­**Опис програми:** для програмної реалізації граничного методу побудови покритів на множинах використано Visual studio code- insader. Програма написана на мові С++.

Приклад роботи програми



**Інструкція користувача(Linux)**

1. Відкриваємо термінал.
2. Переходимо в папку: **cd Public**
3. Компілюємо програму: **g++ -o run Discret1.cpp**
4. Запускаємо програму: **./run**

5. Вводимо дані в термінал.

6. Отримаємо результат

**Лістинг програми**

**#include <iostream>**

**#include <stdio.h>**

**main() {**

**int A[100][100],B[100]= {0},C[100],K[100]= {0},a,b,i,j,k,n,l,e,cina,cina1=20000,kil=0,kp=0,Q[100]= {9},w=10,R[100]= {9},k2,k1;**

**char ch[12]="ABCDEFG ";**

**printf("number of sets: \n");**

**scanf("%d",&a);**

**printf("the number of columns:\n");**

**scanf("%d",&b);**

**printf(" ");**

**for(i=0; i<b; i++) {**

**printf(" ");**

**printf("%d",i+1);**

**}**

**printf(" price");**

**printf("\n");**

**for (i=0; i<a; i++) {**

**printf("\n");**

**printf("%c",ch[i]);**

**for(j=0; j<=b; j++) {**

**printf(" ");**

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

**}**

**}**

**printf("\n");**

**for(e=0; e<a; e++) {**

**cina=0;**

**k=1;**

**l=0;**

**for(i=0; i<b; i++) {**

**C[i]=0;**

**B[i]=0;**

**K[i]=0;**

**}**

**for(j=e; j<a; j++) {**

**C[l]=j;**

**kil++;**

**n=0;**

**for(i=0; i<b; i++) {**

**B[i]+=A[j][i]+K[i];**

**if(B[i]>0) n++;**

**}**

**cina+=A[j][b];**

**if(n!=b) {**

**n=0;**

**for(i=0; i<k; i++)printf("%c",ch[C[i]]);**

**printf(" - \n");**

**for(i=0; i<b; i++) {**

**K[i]=B[i];**

**B[i]=0;**

**}**

**k++;**

**l++;**

**}**

**if(n==b) {**

**n=0;**

**kp++;**

**if(k<w) {**

**for(i=0; i<k; i++)R[i]=C[i];**

**w=k;**

**k2=k;**

**}**

**for(i=0; i<k; i++)printf("%c",ch[C[i]]);**

**if(cina<cina1) {**

**cina1=cina;**

**for(i=0; i<k; i++) Q[i]=C[i];**

**k1=k;**

**}**

**printf("+ price - %d\n",cina);**

**cina=cina-A[j][b];**

**for(i=0; i<b; i++) {**

**B[i]=0;**

**}**

**}**

**}**

**if (C[j]==a) break;**

**}**

**printf("\n\n");**

**printf("answer:\n number of options - %d \nnumber of coatings - %d \n",kil,kp);**

**printf("the cheapest cover - ");**

**for(i=0; i<k1; i++) printf("%c",ch[Q[i]]);**

**printf("\n");**

**printf("the shortest cover - ");**

**for(i=0; i<k2; i++) printf("%c",ch[R[i]]);**

**printf("\n");**

**return 0;**

**}**