**Министерство образования и науки Российской Федерации**

*Федеральное государственное бюджетное образовательное учреждение   
высшего профессионального образования*   
**«АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМ. И.И. ПОЛЗУНОВА» (АлтГТУ)**

Факультет информационных технологий

Кафедра прикладной математики

Отчёт по дисциплине «Алгоритмы и структуры данных»

Лабораторная работа №5

«Динамическое программирование»

Выполнил:

Студент гр. ПИ-32

Глушков Г.Г.

Проверил:

ст. преподаватель

Н.Д. Бубнова

Барнаул 2015

* Формулировка задачи:
* Джин – путешествующий торговец. Он ходит по городам, продавая и покупая товары. Страна, по которой он ходит, состоит из С городов. Начиная свой путь из города S, Джин должен сделать T переходов между городами. Торговцу разрешается посещать города, в которых он уже был. Свой путь Джин должен завершить в одном из E городов. Известна прибыль, которую Джин получает, совершив переход из i - го города в j - ый. Необходимо вычислить максимальную прибыль, которую он может получить, пройдя весь путь
* **Вход.** Состоит из нескольких тестов. Первая строка каждого теста содержит четыре целых числа: количество городов C (2 <= С <= 100), номер начального города S (1 <= S <= 100), число финальных городов E (1 <= E <= 100) и количество переходов T (1 <= T <= 1000). Следующие С строк содержат по С чисел. j - ое число i – ой строки содержит прибыль, которую получит Джин совершив переход из i - го города в j – ый. Поскольку Джин не может идти из i – ого города в i - ый, то i - ое число i – ой строки содержит 0. Далее в одной строке следуют E номеров городов, в которых торговец может завершить свой путь. Последний тест содержит C= S = E = T = 0 и не обрабатывается.
* **Выход.** Для каждого теста вывести максимальную прибыль, которую Джин может получить завершив свой путь.
* Разбор условия: нумерацию городов будем начинать с 0. Обозначим через dk[i] максимальную прибыль, которую может получить торговец, если после совершения k переходов закончит свой путь в городе i (0 <= i < C, города нумеруются от 0 до C - 1). Изначально d0[S] = 0 (нулевая прибыль),d0[i] = -1 (прибыль не определена) для i <= S. Массив fin[i] будет содержать 1, если Джин может завершить свой путь в городе i и 0 иначе. Матрица m[i, j] (0 <= i, j < C) будет содержать прибыль, которую можно получить при переходе из города i в j. Торговец должен совершить T переходов. Для каждого перехода будем пересчитывать значения d[i]. На k - ом переходе в i - ый город можно попасть из j - ого города (0 <= j < C), при этом максимальная прибыль при достижении i - ого города составит dk[i] = (dk-1[j] + m[j, i])Условие dk-1[j] <= 0 гарантирует, что из начального города S за k - 1 переходов можно добраться до города j. Для нахождения результата следует найти максимальное значение среди dT[i], для которых fin[i] = 1 (город i может быть финальным)
* Текст программы с комментариями:

Пока первая строка очередного теста не содержит четыре нуля, читаем

входные данные. Заполняем значения массивов d, fin, m. При этом помним, что нумерация

городов в тестах начинается с 1, а в массивах будем хранить их с 0

#include <stdlib.h>

int main()

{

int c = 0 ;

int s = 0;

int e = 0;

int t = 0;

int \*d ;

int \*d1;

int \*fin;

int i = 0 ;

int j = 0;

while( scanf("%d %d %d %d",&c,&s,&e,&t) ,c + s + e + t > 0)

{

d = (int\*) malloc(c\*sizeof(int));

d1 = (int\*) malloc(c\*sizeof(int));

fin = (int\*) malloc(c\*sizeof(int));

s--;

for( i = 0; i < c; i++)

{d[i] = -1; fin[i] = 0;}

for(d[s] = i = 0; i < c; i++)

{

for( j = 0; j < c; j++)

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

}

for(i = 0; i < e; i++){

scanf("%d",&j); fin[j-1] = 1;

}

/\*Для каждого из T переходов динамически пересчитываем оптимальные значения прибыли

d[i] по выше приведенной формуле. Используем вспомогательный массив d1[i] \*/

while (t > 0)

{

for(i = 0; i < c;i++)

{

for(d1[i] = j = 0; j < c; j++)

{

if (d[j] == -1) continue;

if (d[j] + m[j][i] > d1[i]) d1[i] = d[j] + m[j][i];

}

}

memcpy(d,d1,sizeof(d1)); t--;

}

/\*Ищем максимальное значение d[i], среди тех i, для которых fin[i] = 1. Выводим результат.\*/

for(res = i = 0; i < c; i++)

if ((d[i] > res) && fin[i]) res = d[i];

printf("%d\n",res);

}

* Тесты:



