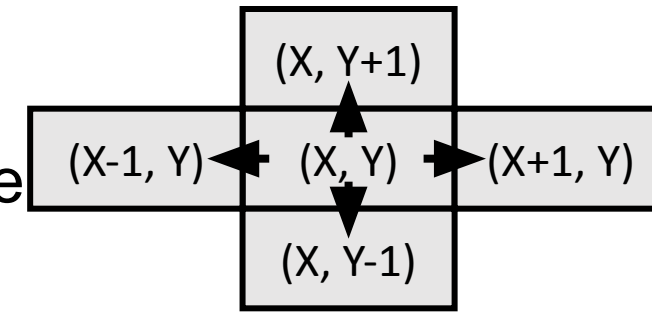


Принципы динамического программирования на примере задачи поиска кратчайшего пути на

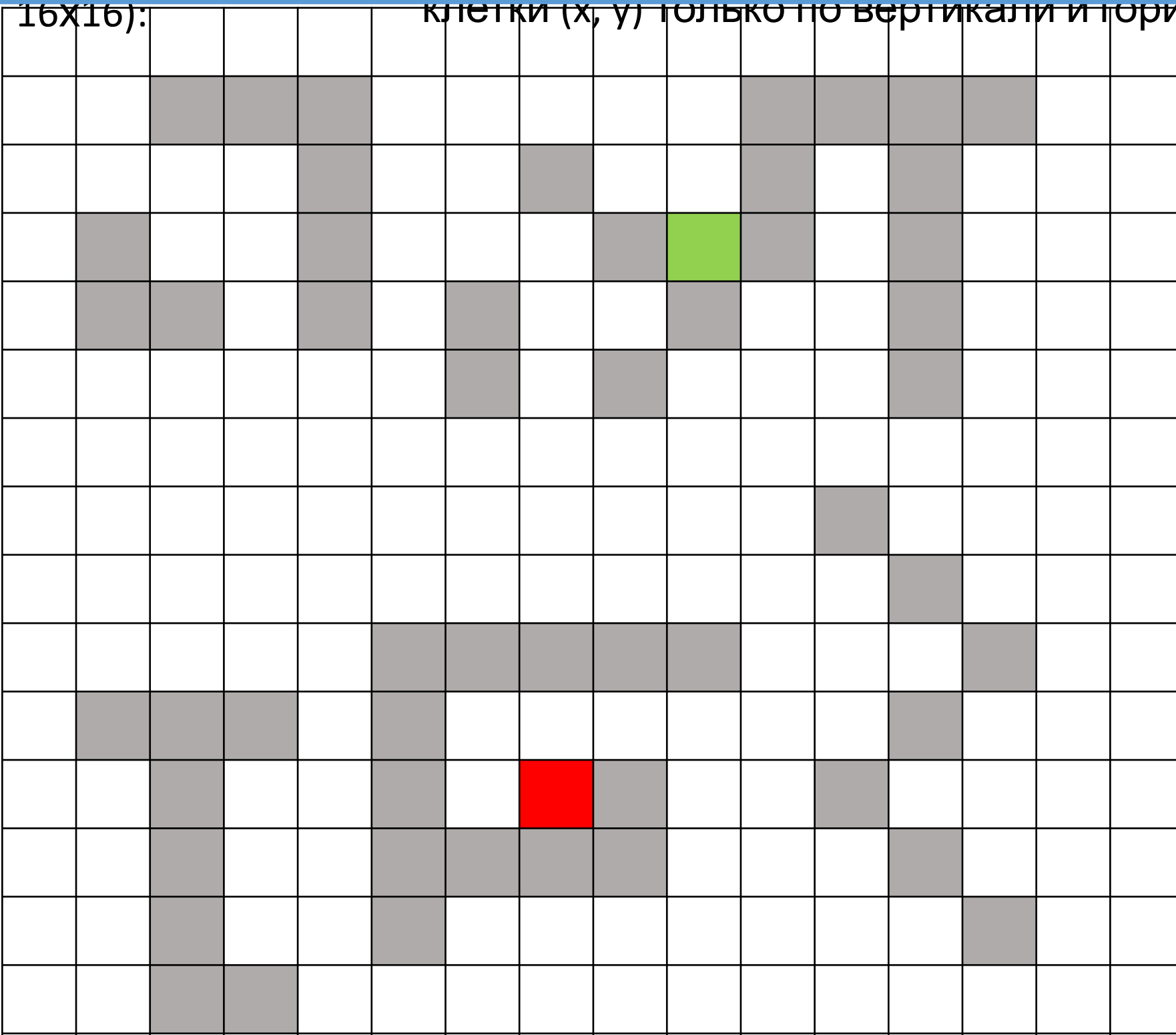
Постановка задачи:

Дано ограниченное поле прямоугольной формы со стенками, где возможно перемещение



Ниже показан пример такого поля (двумерный массив

16x16): клетки (x, y) только по вертикали и горизонтали:



Здесь серые клетки – элементы стен,
 Красная клетка – стартовое поле,
 Зеленая клетка – целевое поле
 Каждую клетку удобно представить как запись со следующими компонентами:

- 1) «Вес» клетки – показывает прибавку к пути, которую дает эта клетка;
- 2,3) Координаты X и Y клетки из которой мы попали в данную клетку;
- 4) Маркер стены – если 1 – через клетку идти нельзя, а если 0, то можно
- 5) Общая длина пути при достижении данной клетки (в клетку идем тогда, когда новый путь короче предыдущего) (все размерности массива увеличим на 2)

Начальную длину пути к каждой клетке удобно задать любым большим числом
 Такая структура позволяет решить

Подсказки к

ДЗ: структура данных (возможна оптимизация):

```
type pp=record
  x,y:integer;//координаты предшественника
  v:integer;//"вес" клетки
  f:integer;//флаг: стенка: f=-1, обычное поле: f=0
  s:integer;//путь до клетки (сумма весов по маршруту)
end;
var pole:array [,] of pp;//двумерный массив записей
    xs,ys,xe,ye,m,n:integer;
```

2) Описание модуля Rread - процедуры подготовки данных (возможна

оптимизация):

```
//открываем файл для чтения;
//читаем из него m,n - количество строк и столбцов поля
setlength(pole,m+2,n+2); //здесь строки поля 1..m, а колонки 1..n. Остальное - стенка
//читаем из файла xs,ys - координаты стартовой клетки (начало координат: [1,1])
//читаем из файла xe,ye - координаты конечной клетки
//читаем в цикле по строкам от 1 до m и столбцам от 1 до n из файла
//веса v всех клеток, и заполняем массив pole, учитывая, что:
// начальное значение S задаем=большое число
// если v=большое число, то это стенка, задаем f=1, иначе f=0
//для нулевой и m+1 - й строки задаем f=1 (стенка)
//для нулевого и n+1 - й колонки задаем f=1 (стенка)
//для клетки (xs,ys) (стартовой) задаем s=0
```

3) Описание модуля Pathfinder(i,j) - процедуры (или функции?) поиска кратчайшего пути из (xs, ys) в (xe,

ye) Пусть (i, j) – текущая клетка, а соответствующий ей элемент массива: pole[i, j].

Если клетка [i+1, j]-не стенка и pole[i, j].s + pole[i+1, j].v < pole[i+1, j].s, тогда:

а) Вычисляем pole[i+1, j].s=pole[i, j].s + pole[i+1, j].v; б) Запускаем Pathfinder(i+1, j)

... и то же самое для остальных направлений

4) Текст (почти полный) ГОЛОВНОЙ

программы:

```
Rread; Pathfinder(xs,ys);
Println('Длина кратчайшего пути из (' ,xs,' , ',ys,') в (' ,xe,' , ',ye,'):',pole[xe,
ye].s);
... Далее идет вывод кратчайшего пути в форме: (xs,ys)->, ..., ->(xe, ye)
```

Буду благодарен за файл с выполненным заданием выслать в мой E-mail адрес: LEOMTL@MAIL.RU, прикрепив к письму и указав в «Теме»