

Семинар 10 по УП

Рекурсия Vol. 2 Търсене с връщане назад



Защо?

- Защото рекурсията е достатъчно сложна за да и отделя още 1 семинар.
- А какво за другия материал?
- Първо трябва важното да усвоите, пък и не остана много материал, който трябва да се покрие!

Малко преговор

- Рекурията представлява функция, която вика сама себе си.
- Дотук добре.
- Още какво знаем? Мии има обратен ход.
- А така. И да осъзнавам, че си говоря и си отговарям сам, просто доде правя тия презентации, достигам ново ниво на шизофрения!
- Ново ниво на наивност също, защото вярвам, че поне някой ще ги отвори вкъщи. Ако е така пишете ми да знам ;)

Продължение на преговора

- Добре какво ни интересува обратния ход?
- Амии можем да кажем какво да стане на правия и какво на обратния ход.
- Как става това?
- Над рекурсивното извикване е правият, а под – обратният.

Пример

```
void rec(int n)
{
    if(n == 0) return;
    std::cout << "Прав ход" << std::endl;
    rec(n-1);
    std::cout << "Обратен ход" << std::endl;
}
```

Това ни помага за задачи

- Можем на правия ход да търсим нещо, а на обратния да променяме или да зануляваме.
- Например задачата с лабиринта от предния семинар. Там на правия ход правим текущата клетка недостъпна и ходим в съседите и, а на обратния ако сме намерили път я сетваме като част от пътя, ако не сме – връщаме старите стойности и търсим другаде.

Обща схема

- Стартираме от някакво начало
- Ако сме във валидно решение просто приключваме и връщаме решението
- Ако това състояние не е валидно решение и няма как от него да стигнем до такова се връщаме и индикираме, че не е.
- Ако състоянието, в което сме не е решение, но има начин да отидем в такова, просто избираме начин и се връщаме в стъпка 2
- Ако избраното състояние не е решение и сме изчерпили всички варианти, просто индикираме, че не е решение и се връщаме.

Задача 1

```
unsigned find_words_in_crossword(char matrix[][255], std::size_t rows, std::size_t cols, char words[][255], std::size_t
```

Напишете функция, която приема кръстословица, представена като матрица от букви, както и списък от думи, и намира броя на думите от списъка, които се срещат в кръстословицата. Една дума се съдържа в кръстословицата, ако всяка буква от думата се намира в съседна клетка на предишната буква. Ако буква в дадена клетка е използвана, тя не може да се ползва втори път.

Пример:

```
char matrix[][255] = {{ 'c', 'a', 't'},  
                      { 'w', 'h', 'o'},  
                      { 'w', 'i', 'n'}};  
  
char words[][255] = {"cat", "hello", "what", "caca"};  
std::cout << find_words_in_crossword(matrix, 3, 3, words, 4); // -> 2  
// -> cat, what
```


Задача 2

Иванчо е започнал работа като разносвач на мляко. Той обикаля през цялото село, като през някои къщи спира, за да вземе мляко, а през някои спира, за да продава мляко. Той обаче може да носи ограничено количество мляко. Един ден, се обадила баба му, която го поканила на гости, но понеже Иванчо е на работа не е сигурен дали ще има възможност да я навести и да се върне обратно. Вашата задача е да напишете програма, която да казва на Иванчо дали ще може да мине през баба си и да се върне обратно в тях. Като вход програмата ще получава число N - размерът на селото, което ще представяме с матрица, която също се получава като вход. Всяка клетка от матрицата представлява къща от селото. Ако в клетка има положително число, то ако Иванчо мине през тази къща трябва да вземе толкова количество мляко, колкото е голямо числото. Ако има отрицателно, то той трябва да продаде толкова количество, колкото е голямо числото по модул. Освен това програмата получава като вход и координатите на къщата на Иванчо, координатите на къщата на баба му и количеството, което Иванчо може да носи. Програмата трябва да проверява дали има път от къщата на Иванчо до къщата на баба му и обратно, като Иванчо трябва да изпълнява поръчките на хората, през чиито къщи минава. Освен това Иванчо иска да е оптимален и да минава само през къщи, през които не е минавал, за да направи най-много оборот. Ако в някакъв момент Иванчо е твърде натоварен или няма достатъчно мляко да даде при посещение на някоя къща, то този път е невалиден.

Задача 3

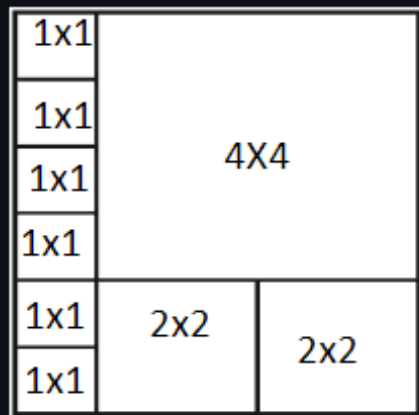
.....

Дадена е шахматна дъска $N \times N$ с конче, поставено върху първия квадрат на празната дъска (1a). Движещият се според правилата на шаха кон трябва да посети всяко поле точно веднъж. Отпечатайте клетките в реда, в който са посетени.

Задача 4

Задача 5 Да се напише рекурсивна функция, която по подадени N и M , размерите на площад. Намерете колко най-малко плочки ще са нужни да запълнят площада, ако плочките са размери $2^n \times 2^n$.

Вход: 5 6 Изход: 9



Задача 5

Задача 3 Да се напише рекурсивна функция, която приема масив и събира всеки две съседни числа, докато не получи 1 число.

Вход: [1, 2, 3, 4, 5] Изход: 48

Поянснение:

[1, 2, 3, 4, 5]

[1 + 2, 2 + 3, 3 + 4, 4 + 5] => [3, 5, 7, 9]

[3 + 5, 5 + 7, 7 + 9] => [8, 12, 16]

[8 + 12, 12 + 16] => [20, 28]

[20 + 28] => 48

Задача 6

- Даден е масив с дължина n . Да се изведат всички подмножества на дадения масив.

Задача 7

- Да се намери най-евтиният път между 2 клетки в матрица, като в клетките с 0 не се стъпва, а цифрата в клетката указва цената за преминаване през нея

Благодаря ви за вниманието!

- Крадени задачи alert.
- Благодарности към Христо, Георги, Вили и Бого за добрият подбор от който си откраднах малко.