

A2, Problem 3 simplification

Posted Oct 9, 2019 7:21 AM

It seems Problem 3 will be difficult for quite a few students. Here is a simpler version of it.

```
int find_path(int n, int m, int A[], int i, int j, int k, int l, int path[][2])
```

```
/*
```

```
    Assume  $n > 0$ ,  $m > 0$ 
```

```
    Assume that (i,j) and (k,l) are within A. That is
```

```
     $i \geq 0 \ \&\& \ i < n \ \&\&$ 
```

```
     $k \geq 0 \ \&\& \ k < n \ \&\&$ 
```

```
     $j \geq 0 \ \&\& \ j < m \ \&\&$ 
```

```
     $l \geq 0 \ \&\& \ l < m$ 
```

```
    To make it a bit easier, you can use extra array(s).
```

```
    (10 points) If there is no path between (i,j) and (k,l), return 0
```

```
    If there is a path return  $> 0$ . The return value does not have to be the length of a path. If (i,j) and (k,l) are the same,  $i == k \ \&\& \ j == l$ , and both contain 0, return 1
```

```
    Bonus (5 points)
```

```
    If there is a path between (i,j) and (k,l) return its length and the path in path[][2].
```

```
    If (i,j) and (k,l) are the same,  $i == k \ \&\& \ j == l$  and both are 0, return length 1 and path[0][0] = i, path[0][1] = j
```

```
*/
```