# CPS 130 Homework 14
# Dynamic Programming

*Write and justify your answers in the space provided.*[1]

1. A game-board consists of a row of $n$ fields, each consisting of two numbers. The first number can be any positive integer, while the second is 1, 2, or 3. An example of a board with $n = 6$ could be the following:

| 17 | 2 | 100 | 87 | 33 | 14 |
|---|---|---|---|---|---|
| 1 | 2 | 3 | 1 | 1 | 1 |

   The object of the game is to jump from the first to the last field in the row. The top number of a field is the cost of visiting that field. The bottom number is the maximal number of fields one is allowed to jump to the right from the field. The cost of a game is the sum of the costs of the visited fields.

   Let the board be represented in a two-dimensional array $B[n, 2]$. The following recursive procedure (when called with argument 1) computes the cost of the cheapest game:

```
Cheap(i)

     IF i>n THEN return 0
     x=B[i,1]+Cheap(i+1)
     y=B[i,1]+Cheap(i+2)
     z=B[i,1]+Cheap(i+3)
     IF B[i,2]=1 THEN return x
     IF B[i,2]=2 THEN return min(x,y)
     IF B[i,2]=3 THEN return min(x,y,z)

END Cheap
```

[1] Collaboration is allowed, even encouraged, provided that the names of the collaborators are listed along with the solutions. Students must write up the solutions on their own.

(a) Analyze the asymptotic running time of the procedure.

(b) Describe and analyze a more efficient algorithm for finding the cheapest game.