CPS 130 Homework 14 - Solutions

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
```

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

Solution:

$$\begin{aligned} T(n) &= T(n-1) + T(n-2) + T(n-3) + \Theta(1) \\ &\geq 3\,T(n-3) \\ &= 3^2\,T(n-6) \\ &= \ldots \\ &= 3^k\,T(n-3k) \\ &\geq 3^n \\ &= \Omega(3^n). \end{aligned}$$

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

Solution: We create a table T of size n in which to store our results of prior runs. The modified algorithm would be as follows:

```
Cheap(i)

    IF T[i] != ∅ THEN return T[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 T[i] = x
    IF B[i,2]=2 THEN T[i] = min(x,y)
    IF B[i,2]=3 THEN T[i] = min(x,y,z)
    return T[i]

END Cheap
```

The cost of a recursive call is $O(1)$ and we fill each entry in the table at most once, so the running time is $O(n)$.