Zach Domke

5/23/18

CIS 315

Chris Wilson

1. 2010 = ((m1 ∗ m2)((m3 ∗ m4)(m5 ∗ m6)))
2.
   a. At each second we are to determine whether it is more beneficial to use our EMP now and destroy f(j) robots or if we should let it charge to destroy f(j+1) robots.
   b. $$robots[i, j] = \begin{cases} 0 & \text{if } i = j \\ f(i-j) & \text{if } i > j \end{cases}$$

$$sum[j] = \begin{cases} 0 & \text{if } j = 0 \\ \min\{sum[i-1] + robots[i, j] \mid 0<i<=j\} & \text{if } j > 0 \end{cases}$$

   c.
   d. time is $O(n^2)$ and size is $O(n^2)$
3.
   a. We want to find the minimum cost and assign that to each line. First we find the cost of a line that is possible using provided variables i and j by applying them to the sum provided (which we will call sum[i, j]). Next, we apply the cost of a line with variables i and j by cubing the potential extra spaces (called penalty(i, j)). Finally, we attempt to find the minimum cost to apply to each line (called cost(j)).
   b. $$penalty[i, j] = \begin{cases} \infty & \text{if } sum[i, j] < 0 \\ 0 & \text{if } j = n \ \&\ sum[i, j] >= 0 \\ (sum[i, j])^3 & \text{else} \end{cases}$$

$$cost[j] = \begin{cases} 0 & \text{if } j = 0 \\ \{\min\{cost[i-1] + penalty[i, j] \mid 0<i<=j\} & \text{if } j > 0 \end{cases}$$

c. foo()

```
sum[][], penalty[][], cost[]
for 0 < i <= n
     sum[i, i] = M − l_i
     for i < j <= n
          sum[i, j] = sum[i, j−1] − l_j − 1
for 0 < i <= n
     for i <= j < n
          if sum[i, j] < 0
               penalty[i, j] = ∞
          elif j == n && sum[i, j] >= 0
               penalty[i, j] = 0
          else
               penalty[i, j] = (sum[i, j])^3
cost[0] = 0
for 0 < j < n
     cost[j] = ∞
     for 0 < i < j
          if cost[i−1] + penalty[i, j] < cost[j]
               cost[j] = cost[i−1] + penalty[i, j]
return cost[n]
```

d. time is $O(n^2)$ and space is $O(n^2)$

4. foo()

```
C[] = ∞ from C[−T] to C[T]
temp[] = ∞ from temp[0] to temp[n−1]
coins[] = ∞ from coins[0] to coins[T]
C[0] = 0
for 0 < i < T
     for 0 <= j < n
          place = t−d_j
          if place >= 0
               temp[j] = C[place]
```

```
     x = 0
     for 0 <= j < n
          if temp[j] < temp[x]
               x = j
     C[i] = 1 + temp[x]
     coins[i] = d_x
for 0 <= i < T
     if coins[i] != ∞
          print coins[i]
print C[T]
return
```