Group Assignment #2

Kyle Nichols, Alex Bailey, Justin Pham

**Psuedocode**

The following is pseudocode for the enumerative and dynamic programming algorithms.

*Algorithm 1:*

*Algorithm 2:*

MinLockers(balls[1,...,n], keys[1,...,n], n, B, K):

int d[B]

d[0] = 0;

for i = 1 to N

if balls[i] == 0 // There is no ball

d[i] = d[i-1]

else if balls[i] == 1 and keys[i] == 0 // There is a ball, but we don't have the key

distance = Dist(balls, keys, i, n) // How many lockers need to be opened to get to the key

d[i] = 1 + distance + d[i-distance]

else if balls[i] == 1 and keys[i] == 1 // There is a ball and we have the key

d[i] = 1 + d[i-1]

return d[n]

Dist(balls[1,...,n], keys[1,...,n], i, n):

// Search above the ball location

for j = i+1 to n-1

if keys[j] == 1

distance = j - i

if distance < minDistance

minDistance = distance

firstLoop = true

secondLoop = false

// Search below the ball location

for k = i-1 to 0

if keys[k] == 1

distance = i - k

if distance < minDistance

minDistance = distance

firstLoop = false

secondLoop = true

if firstLoop

for k = i to i+minDistance

keys[k-1] = 1

keys[k+1] = 1

else if secondLoop

for j = i-minDistance to i

keys[j-1] = 1

keys[j+1] = 1

return minDistance

**Analysis**

The following is a runtime analysis for the enumerative and dynamic programming algorithms.

*Algorithm 1:*

*Algorithm 2:*

The lower bound for the algorithm is likely Ω(M), as the best case scenario is that your all the tennis balls are matched 1:1 with a given key. The worst case runtime is when all of the loops are run through completely. In our implementation of this, there is an outer loop that runs iteratively from the first locker to the last, and an inner set of for loops will do the same thing (though likely never will), followed by a smaller loop. This gives N(N+cN), with c being some constant. Ultimately, our upperbound is Ο(N2 + cN2) and Θ(N2).

**Solutions**

The following is a list of solutions for the input sequences provided for the enumerative and dynamic programming algorithms.

*Algorithm 1:*

*Algorithm 2:*

|  |  |
| --- | --- |
| Test | Output |
| 1 | 191 |
| 2 | 23 |
| 3 | 89 |
| 4 | 31 |
| 5 | 146 |
| 6 | 33 |
| 7 | 87 |
| 8 | 85 |