Project 2 Exhaustive vs Dynamic Crane Problem

Connor Bither

[John.bither@csu.fullerton.edu](mailto:John.bither@csu.fullerton.edu)

Exhaustive Search:

Pseudocode

best = path(setting)

max\_steps = setting.get\_max\_steps()

for steps from 0 to max\_steps:

for parts from 0 to 2^steps - 1:

potential = path(setting)

for i from 0 to steps:

part = (parts >> i) & 1

if part == 1 and potential.is\_step\_valid(STEP\_DIRECTION\_EAST):

potential.add\_step(STEP\_DIRECTION\_EAST)

else if potential.is\_step\_valid(STEP\_DIRECTION\_SOUTH):

potential.add\_step(STEP\_DIRECTION\_SOUTH)

if best.last\_step() == STEP\_DIRECTION\_START or potential.total\_cranes() > best.total\_cranes():

best = potential

return best

Time Complexity – O(2^n)

Dynamic:

Pseudocode

A = [[None for j in range(c)] for i in range(r)]

A[0][0] = [(0, 0)] # base case

for i in range(r):

for j in range(c):

if G[i][j] == X:

A[i][j] = None

else:

from\_above = None

if i > 0 and G[i-1][j] != X and A[i-1][j] is not None:

from\_above = A[i-1][j] + [(i, j)]

from\_left = None

if j > 0 and G[i][j-1] != X and A[i][j-1] is not None:

from\_left = A[i][j-1] + [(i, j)]

if from\_above is None and from\_left is None:

A[i][j] = None

elif from\_above is None:

A[i][j] = from\_left

elif from\_left is None:

A[i][j] = from\_above

else:

if len(from\_above) > len(from\_left):

A[i][j] = from\_above

else:

A[i][j] = from\_left

max\_cranes = 0

max\_path = None

for i in range(r):

for j in range(c):

if A[i][j] is not None and len(set(A[i][j])) == len(A[i][j]) and len(A[i][j]) > max\_cranes:

max\_cranes = len(A[i][j])

max\_path = A[i][j]

return max\_path

Time Complexity – O(n^3)

Graphs

A picture containing line, plot, diagram, parallel

Description automatically generated

Red – exhaustive search

Blue – dynamic

Questions

1. Is there a noticeable difference in the performance of the two algorithms? Which is faster, and by how much? Does this surprise you?

The Dynamic program was noticeably faster which did kind of surprise me as at low numbers of inputs that we are working with I thought they would be close.

1. Are your empirical analyses consistent with your mathematical analyses? Justify your answer.

My empirical analyses were that the dynamic program was consistently faster which matches what my mathematical analyses expected.

1. Is this evidence consistent or inconsistent with hypothesis 1? Justify your answer.

The evidence is consistent with hypothesis 1 because for the worst case scenarios exhaustive search is greater than that of dynamic programming and as such will generally take longer

1. Is this evidence consistent or inconsistent with hypothesis 2? Justify your answer.

The evidence is inconsistent with the second hypothesis as while the dynamic program can be worse than the exhaustive search at times, for the majority of the time it will be more efficient.