Project 2: Dynamic vs. Exhaustive - Crane unloading problem

CPSC 335 - Algorithm Engineering

Spring 2023

By: Jair De Orta jairdeorta@csu.fullerton.edu

Exhaustive Algorithm Pseudocode:

```
best path          1tu
for k=0 to (2^max steps-1) do     2^n tu
        new path x        1tu

        for i=0 to max_steps-1 do   n tu
                bit = k >> i & 1 3tu
                if bit = 0 1 tu
                        Step Direction East
                else
                        Step Direction South
        for direction do  p tu
                if the step is not valid(building) 1tu
                        break
                else
                        add step 1tu

                if path x total cranes >= best total cranes  1tu
                        best = x            1tu
return best
```

1+ 2^n*(1+4n+4p)

p is a constant so it's mostly 2^n*4n+constant

so time complexity is O(2^n*n)

Dynamic Algorithm Pseudocode:

A[0][0] = path 1tu

check A[0][0] has value

for coordinate r to rows-1 do n tu

for coordinate c to columns-1 do n tu

if [r][c] is cell building 1tu

A[r][c] resets

continue

cell type above 1tu

cell type left 1tu

if r>0 & [r-1][c] is not building & [r-1][c] has value 4tu

insert A[r-1][c] value and south to above

if c>0 & [r][c-1] is not building & [r][c-1] has value 4tu

insert and add A[r][c-1] value and east to left

if above and left have value 1tu

if left has more cranes than above 1tu

A[r][c] = left 1tu

else

A[r][c] = above 1tu

else if above has value

A[r][c] = above 1tu

else if left has value

A[r][c] = left 1tu

cell type best = [0][0] 1tu

for coordinate r to rows-1 do n tu

for coordinate c to columns-1 do ntu

if A[r][c] total cranes > best total cranes 1tu

best = A[r][c] 1tu

return best

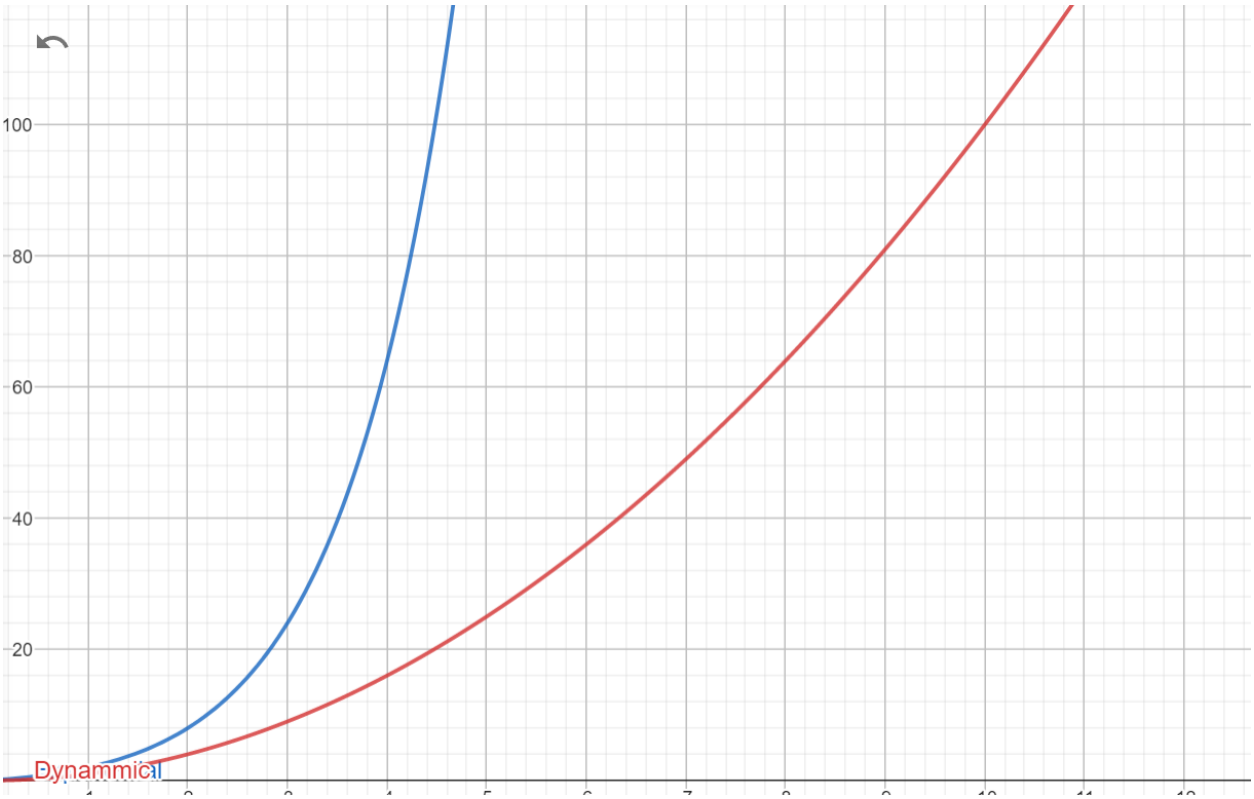
1+n*(16n)+1+n*(2n)

16n^2 + 2n^2 +2

=18n^2+2

Time complexity is O(n^2)

Graph:



Questions

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

Yes, there is a noticeable difference between the two algorithms. The dynamic programming algorithm(red) is faster than the exponential algorithm(blue) and by a decent margin. Looking at both time complexities logically then it does make sense, so I was surprised initially but then, thinking about it, it makes sense.

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

Yes, the empirical analyses are consistent with the mathematical analyses. Just by looking at the graph and how steep exponential is compared to dynamic, it shows the difference when input size increases and how much longer the exponential algorithm will take. And viewing mathematically, it makes sense. As n, increases the answer for exponential becomes much bigger than the dynamic's algorithm.

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

The evidence is consistent with polynomial-time dynamic programming algorithms being more efficient than exponential-time exhaustive search algorithms that solve the same problem. With the graph and even just testing the time complexities mathematically, overall, the dynamic algorithm will be more efficient. Being much faster with more input size.

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

No, it is not. With evidence pointing toward dynamic algorithm being more efficient, the 2nd hypothesis is inconsistent.