Alex Merino

2/4/2023

Professor Chan

CSE 2010

HW 2

1. The number of additions to the total would be the ceiling of (n/2) since i is incrementing by 2 each time. And if n is an odd number such as 5 the for loop would run 3 times because on the second loop i = 4 and since it is still less than 5 it will run a 3rd time.
2. The number of additions to the total would be $\frac{(n+1)(n+2)}{2}$ because it is very similar to the summation of n except for the fact that since j >= 0, there will always be an extra iteration where the total is added, so if n = 5, when i = 5 the nested loop would run 6 times which when plugged in gives us the number of additions.
3. $If\ d(n) = n + 1 \qquad n + 1\ \leq c'n$

   $If\ e(n) = n + 2 \qquad n + 2\ \leq c''n$

   $(n + 1) + (n + 2) \leq\ c'n + c''n$

   $2n + 3 \leq\ n(c' + c'')$

   $d(n) + e(n)\ is\ O(f(n) + g(n))$
4. $3n^2 + 4n - 3\ \leq cn^2$

   $4n - 3\ \leq cn^2 -\ 3n^2$

   $4n - 3\ \leq n^2(c - 3)$

$$\frac{4n-3}{(c-3)} \leq n^2$$

$$4n-3 \leq n^2 \quad if\ c=4$$

$$-3 \leq n^2 - 4n \ \ if\ c = 4$$

$$-3 \leq n(n-4) \ \ if\ c=4$$

$$if\ n_0 = 5\ and\ \ c = 4\ then\ O\big(f(n^2)\big)\ is\ true$$

$$3n^2 + 4n - 3 \geq\ cn^2$$

$$3n^2 + 4n - 3 \geq\ 3n^2 \quad c = 3$$

$$4n - 3 \geq 0 \quad c = 3$$

$$if\ n_0 = 1\ and\ c = 3\ then\ \Omega(f(n^2))$$

$$c'n^2 \leq 3n^2 + 4n - 3\ \leq c''n^2$$

$$3n^2 \leq 3n^2 + 4n - 3\ \leq 4n^2\ \ c' = 3 \ \ c'' = 4$$

$$if\ n_0 = 5 \ \ c' = 3\ c'' = 4\ then\ \theta\big(f(n^2)\big)\ is\ true$$

5. T(1) = 1 because there is only one element in the array which means that the key is either in the array or not. T(2) = 2 because both mid variables calculated would be 0 and if it is not in the 0th position it is in the first position or not in the array at all. T(n) = 2 + T(n/3) characterizes tertiary search because for every n > 2 it would take at the least n/3 to narrow down the array plus 2 more once the sub-array gets down to only two items in case the key is not in the array at all.

$$T(n) = 2 + T(\frac{n}{3})$$

$T(n) = 4 + T(\frac{n}{9})$

$T(n) = 6 + T(\frac{n}{27})$

$T(n) = 2k + T(\frac{n}{3^k})$

$T\left(\frac{n}{3^k}\right) gets\ smaller\ until\ T(1)\ so\ 3^k = n \quad k = \ log_3 n$

$T(n) = 2log_3 N + T(\frac{n}{n})$

$T(n) = 2log_3 N + T(\frac{n}{n})$

$T(n) = 2log_3 N + 1$

$O(log_3 n)\ algorithm$

6.

a. (n-1)! is the number of all possible routes in this problem since the first station has to be the PackingStation. The rest of the stations have to be organized in everyway possible which is why the factorial is needed.

b. If n = 100 then 99! is about $9.33\ x\ 10^{155}$ and 1 billion is $1\ x\ 10^9$ then it would take $9.33\ x\ 10^{146}\ seconds$ for the program to finish. Converting seconds to years by dividing by 31,536,000 which is how many seconds in a year. The amount of years it would take would be $2.96\ x\ 10^{139}\ years$ which is still incomprehensively large. The universe is only $13.7\ x\ 10^9$ years old which means that the program would still need to be run $2.16\ x\ 10^{130}$ times the age of the universe.

c. The computer can run 60 billion routes in one minute so the number of stations that can be used is whichever integer n-1 whose factorial is less than 60 billion.

The number n would be 14 since (14-1)! is the closest to 60 billion without going over.