Noah Buchanan
Problem Set 1
Algorithms

August 21, 2020

1. **Base Case:**

   suppose n = 1

$$\begin{aligned} 2n - 1 &= n^2 \\ 2(1) - 1 &= 1^2 \\ 1 &= 1 \end{aligned}$$

   $\therefore$ P(1) is true

   Inductive step:

   Let n = k
   Let k $\geq$ 1
   Assume P(k) is true

$$\begin{aligned} 1 + 3 + 5 + ... + (2k-1) + (2(k+1)-1) &= (k+1)^2 \\ k^2 + 2k + 1 &= (k+1)^2 \\ (k+1)(k+1) &= (k+1)^2 \end{aligned}$$

   $\therefore$ P(n+1) is true

   $\therefore$ $\forall$ n $\geq$ 1, If P(n) is true, then P(n+1) is also true; P(n) $\longrightarrow$ P(n+1)

   $\therefore$ $\forall$ n $\geq$ 1, P(n) is true

2. **Base Case:**

   suppose n = 1

$$
\begin{aligned}
n^2 &= \frac{n(n+1)(2n+1)}{6} \\
1 &= \frac{1(2)(3)}{6} \\
1 &= \frac{6}{6}
\end{aligned}
$$

   $\therefore$ P(1) is true

   Inductive step:

   Let n = k
   Let k $\geq$ 1
   Assume P(k) is true

$$
\begin{aligned}
1^2 + 2^2 + 3^2 + ...k^2 + (k+1)^2 &= \frac{(k+1)((k+1)+1)(2(k+1)+1)}{6} \\
\frac{k(k+1)(2k+1)}{6} + \frac{6(k+1)^2}{6} &= \frac{(k+1)(k+2)(2k+3)}{6} \\
k(k+1)(2k+1) + 6(k+1)^2 &= (k+1)(k+2)(2k+3) \\
2k^3 + 9k^2 + 13k + 6 &= 2k^3 + 9k^2 + 13k + 6
\end{aligned}
$$

   $\therefore$ P(n+1) is true

   $\therefore$ $\forall$ n $\geq$ 1, If P(n) is true, then P(n+1) is also true; P(n) $\longrightarrow$ P(n+1)

   $\therefore$ $\forall$ n $\geq$ 1, P(n) is true

3. (a) **Initialization:**

   Suppose n = 0, Before we even begin looping, (which it will not since n = 0), i and p are both 1 and currently p = k^0 so it holds true at this stage.

   (b) **Maintenance:**

   suppose n = n+1, 1 * k = k and k^1 = k so currently p is still equal to k^n.

(c) **Termination:**

p which is equal to 1, has now been multiplied by k, n number of times so p at termination will also equal k^n

4. (a) **Initialization:**

   We start at index 0 of A[] and if A[i] = k (the number that we are looking for we return the value, otherwise we start back at the top of the loop.

   (b) **Maintenance:**

   Increment i and check if the new index we are on is the value we are searching for, if it is return the value, if it is not go to the top of the loop and check if the conditions are ok to begin another loop and is done until we find the value or we are at the end and have not found it.

   (c) **Termination:**

   The value in question has either been found and we return i, or we are at the end of the array and the last value in A[] was not the value we are searching for so we return null.

5. The average number of elements to be compared considering that all values in the array have equal probabilities to be the key is n/2, this is based on the fact that any range of numbers ranged 1 to n must always have two extremes for comparing 1 and n compares, the overall average you could technically say of 1 to n/2 and n/2 to n are both 50 percent chance if it is evenly distributed.

6. for i = 1 to n executes lgn + 1 times

   $T(n) = (c_1 * (lgn + 1))$

   i = n/2 executes lgn times

   $T(n) = (c_1 * (lgn + 1)) + (c_2 * (lgn))$

   while i > 0 executes ($\frac{n(n+1)}{2}$+1) * lgn times

   $T(n) = (c_1 * (lgn + 1)) + (c_2 * (lgn)) + (c_3 * (\frac{n(n+1)}{2} + 1) * lgn)$

   Print(i) and i = i - 1 both execute ($\frac{n(n+1)}{2}$) * lgn times

   $T(n) = (c_1 * (lgn + 1)) + (c_2 * (lgn)) + (c_3 * (\frac{n(n+1)}{2} + 1) * lgn) + (c_4 * (\frac{n(n+1)}{2}) * lgn) + (c_5 * (\frac{n(n+1)}{2}) * lgn)$

   k = 1 executes lgn times

$T(n) = (c_1 * (lgn + 1)) + (c_2 * (lgn)) + (c_3 * (\frac{n(n+1)}{2} + 1) * lgn) + (c_4 * (\frac{n(n+1)}{2}) * lgn) + (c_5 * (\frac{n(n+1)}{2}) * lgn) + (c_6 * lgn)$

while k < n executes ($\frac{n(n+1)}{2} + 1$) * lgn times

$T(n) = (c_1 * (lgn + 1)) + (c_2 * (lgn)) + (c_3 * \frac{n(n+1)}{2} + 1) + (c_4 * \frac{n(n+1)}{2}) + (c_5 * \frac{n(n+1)}{2}) + (c_6 * lgn) + (c_7 * (\frac{n(n+1)}{2} + 1) * lgn)$

k = k+1 executes ($\frac{n(n+1)}{2}$) * lgn times $T(n) = (c_1 * (lgn + 1)) + (c_2 * (lgn)) + (c_3 * \frac{n(n+1)}{2} + 1) + (c_4 * \frac{n(n+1)}{2}) + (c_5 * \frac{n(n+1)}{2}) + (c_6 * lgn) + (c_7 * (\frac{n(n+1)}{2} + 1) * lgn) + (c_8 * (\frac{n(n+1)}{2}) * lgn)$

Most simplified form:

$T(n) = n^2(\frac{c3}{2} + \frac{c4}{2} + \frac{c5}{2} + lgn(\frac{c7}{2} + \frac{c8}{2})) + n(\frac{c3}{2} + \frac{c4}{2} + \frac{c5}{2} + lgn(\frac{c7}{2} + \frac{c8}{2})) + lgn(c1 + c2 + c6 + c7) + (c1 + c3)$