Homework 1

Lucas Machado Moschen

1 Induction

Answers should be written in this document.

1. Prove by Induction that: $\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6} \qquad \forall n \geq 0$

   Resposta: Considere o caso base $n = 1$. $\sum_{i=1}^{1} i^2 = 1 = \frac{1*2*3}{6} = 1$, portanto a propriedade vale para $n = 1$. Agora suponha que $\sum_{i=1}^{k} i^2 = \frac{k(k+1)(2k+1)}{6}$ é válido. Devo provar que esse somatório vale para $k + 1$.

$$\sum_{i=1}^{k+1} i^2 = \sum_{i+1}^{k} i^2 + (k+1)^2 = \frac{k(k+1)(2k+1)}{6} + (k+1)^2$$

   Esse resultado vem da hipótese de indução. Fazendo as operações algébricas, obtemos:

$$\begin{aligned}\frac{k(k+1)(2k+1)}{6} + (k+1)^2 &= \frac{k(k+1)(2k+1) + 6(k+1)^2}{6} \\ &= \frac{(k+1)[k(2k+1) + 6(k+1)]}{6} \\ &= \frac{(k+1)(2k^2 + 7k + 6)}{6} \\ &= \frac{(k+1)((k+1)+1)(2(k+1)+1)}{6}\end{aligned}$$

   Desta maneira, pelo Princípio de Indução, a igualdade é válida.

2. Prove by Induction that: $\forall n \geq 7$ it is true $3^n < n!$

   Resposta: Seja $P(n) : 3^n < n!$ a propriedade a ser demosntrada por indução. O caso base é para $n = 7$:

$$P(7) : 3^7 = 2187 < 5040$$

Agora suponha que $P(k)$ é válida. Queremos provar que $P(k) \Rightarrow P(k+1)$. Assim:

$$3^{k+1} = 3^k \cdot 3^1 < k! \cdot 3 < k! \cdot (k+1) = (k+1)! \implies 3^{k+1} < (k+1)!$$

Logo, pelo Princípio de Indução, a propriedade $P(n)$ é válida. Observe que a primeira desigualdade utilizou a hipótese de indução. Também observe que a segunda desigualdade usa a ideia de que $3 < k$, pois $k \geq 7$.

3. Prove by Induction that $\forall n \geq 0$

$$\left\lceil \frac{n}{2} \right\rceil = \begin{cases} \frac{n}{2} & \text{si } n \text{ es par} \\ \frac{n+1}{2} & \text{si } n \text{ es impar} \end{cases}$$

Resposta: Considere a propriedade $P(n)$ a que se segue acima. É fácil ver que 0 é par e, portanto:

$$P(0) : \lceil \frac{0}{2} \rceil = \lceil 0 \rceil = 0 = \frac{0}{2}$$

Logo $P(0)$ é válida.

Suponha $P(k)$ válida. Quero mostrar que $P(k) \Rightarrow P(k+1)$. Agora, considere os casos:

- k é par. $\lceil \frac{k}{2} \rceil = \frac{k}{2}$. Logo, $\frac{k}{2} < \frac{k+1}{2} \leq \frac{k}{2} + 1$. $\frac{k}{2}$ é um número inteiro e por definição de função teto:

  $$P(k+1) : \lceil \frac{k+1}{2} \rceil = \frac{k}{2} + 1 = \frac{(k+1)+1}{2}$$

  Logo $P(k+1)$ é válida para k par.

- k é ímpar. Nesse caso, $k+1$ é par e $\frac{k+1}{2}$ é um número inteiro. Desta maneira, $\frac{k+1}{2} - 1 < \frac{k+1}{2} \leq \frac{k+1}{2}$ e, por definição de função teto:

  $$P(k+1) : \lceil \frac{k+1}{2} \rceil = \frac{k+1}{2}$$

  Logo $P(k+1)$ é válida para k ímpar.

  Conclui-se que, pelo Princípio de Indução, $P(n)$ é verdadeira.

4. Prove by induction that a number is divisible by 3 if and only if the sum of its digits is divisible by 3.

Resposta: ($\Rightarrow$) Seja $P(n)$: a soma dos dígitos de $3n$ é múltiplo de 3.

$P(0)$: a soma dos dígitos de $3 \cdot 0 = 0$ é 0, logo, é múltiplo de 3. O caso base é válido. Suponha $P(k)$ válida. $3k = a_m a_{m-1} a_{m-2} ... a_2 a_1$, sendo a_i dígito de $3k$, tal que $a_m + a_{m-1} + ... + a_2 + a_1$ é múltiplo de 3. Seja $P(k+1)$. Considere os casos:

- $a_1 \leq 6$. $3(k+1) = a_m a_{m-1} a_{m-2} ... a_2(a_1+3)$ e $a_m + a_{m-1} + ... + a_2 + a_1 + 3$, a soma dos dígitos de $3k+3$, é múltiplo de 3.
- $9 \geq a_1 > 6$. $3k+3 = a_m a_{m-1} a_{m-2} ...(a_2+1)(a_1-7)$ e $a_m + a_{m-1} + ... + a_2 + 1 + a_1 - 7 = a_n + ... + a_1 - 6$, que é múltiplo de 3.

Assim, pelo Princípio de Indução, $P(n)$ é verdadeira. A prova para os número inteiros não positivos é análoga.

($\Leftarrow$) Seja $Q(n)$: se um número x tem n dígitos tal que a soma deles é múltiplo de 3, então $3|x$. $Q(1)$ é válida, pois a soma de seus dígitos é si próprio, logo, é múltiplo de 3. Suponha válida $Q(k)$, hipótese de indução. Considere $x = a_{k+1} a_k a_{k-1} ... a_3 a_2 a_1$. Separe esse número em $x_1 = a_1$ e $x_2 = a_{k+1} a_k ... a_3 a_2$. Considere os casos:

- $a_1 \equiv 0 \mod 3$: Nesse caso, $a_{k+1} + a_k + ... + a_2 \equiv 0 \mod 3$ e, por hipótese, $3|x_2 \Rightarrow 3|x_2 \cdot 10 \Rightarrow 3|x_2 \cdot 10 + x_1 = x$, e está provado.
- $a_1 \equiv 1 \mod 3$: $a_{k+1} + a_k + ... + a_2 \equiv 2 \mod 3$. Tomo $a_1 - 1$ Aqui, separemos em dois casos:
  - $a_2 = 9$: Tomo $a_2 - 2$. O número $a_{k+1} a_k ... a_3(a_2-2)$ é múltiplo de 3 por hipótese. Assim $a_{k+1} a_k ... a_3(a_2-2) \cdot 10 + a_1 - 1 = a_{k+1} a_k ... a_3 a_2 \cdot 10 + a_1 - 20 - 1$ também será múltiplo de 3. Como $3|-21$, então $3|a_{k+1} a_k ... a_3 a_2 \cdot 10 + a_1 = a_{k+1} a_k ... a_3 a_2 a_1 = x$ e está provada.
  - $a_2 < 9$: Tomo $a_2 + 1$. O número $a_{k+1} a_k ... a_3(a_2+1)$ é múltiplo de 3 por hipótese. Assim $a_{k+1} a_k ... a_3(a_2+1) \cdot 10 + a_1 - 1 = a_{k+1} a_k ... a_3 a_2 \cdot 10 + a_1 + 10 - 1$ também será múltiplo de 3. Como $3|9$, então $3|a_{k+1} a_k ... a_3 a_2 \cdot 10 + a_1 = a_{k+1} a_k ... a_3 a_2 a_1 = x$ e está provada.
- $a_1 \equiv 2 \mod 3$: análogo ao item anterior, alterando os casos para $a_2 = 0$, em que você soma 2 e $a_2 > 0$, em que se diminui 1. Por fim, os casos serão similares e estará provado.

Conclui-se que $Q(k) \Rightarrow Q(k+1)$, logo $Q(n)$ é verdadeira.

5. Prove that any integer greater than 59 can be formed using only 7 and 11 cent coins.

Resposta: Seja $P(n)$: $n = 7m + 11n$, para algum m e algum n inteiros. Provemos o caso inicial:

$$P(60) : 60 = 7 \cdot 7 + 11 \cdot 1$$

$P(60)$ é válida. Suponha válida $P(k) : k = 7m + 11n$. Assim, $P(k+1) : k+1 = 7m + 11n + 1$. Considere os casos:

- $m \geq 3$: Nesse caso, $k+1 = 7m + 11n + 11 \cdot 2 - 7 \cdot 3 = 7(m-3) + 11(n+2)$, portanto é verdade.
- $m < 3$: Nesse caso, sabemos que $n > 4$, pois se $n \leq 4$, $7m < 21 \rightarrow 7m + 11n \leq 14 + 11n \leq 14 + 44 = 58 < 60$, que é o mínimo que desejamos. Assim, $k+1 = 7m + 11n + 7 \cdot 8 - 11 \cdot 5 = 7(m+8) + 11(n-5)$

Conclui-se, pelo Princípio de Indução, que $P(n)$ é válida, desde que $n > 59$.

6. Prove by induction that $F_{n+k} = F_k F_{n+1} + F_{k-1} F_n$

Resposta: Seja $k > 1$ uma constante e façamos a indução sobre n. Provemos os casos iniciais:

$$F_{0+k} = F_k F_{0+1} + F_{k-1} F_0 \Rightarrow F_k = F_k F_1 + F_{k-1} F_0 \Rightarrow F_k = F_k$$

$$F_{1+k} = F_k F_{1+1} + F_{k-1} F_1 \Rightarrow F_{k+1} = F_k F_2 + F_{k-1} F_1 \Rightarrow F_{k+1} = F_k + F_{k-1}$$

Note que foi usado o fato de $F_0 = 0, F_1 = 1, F_2 = 1$ e $F_k = F_{k-1} + F_{k-2}$. Assim, está provado. Suponha agora que $F_{n+k} = F_k F_{n+1} + F_{k-1} F_n$ e que $F_{n+1+k} = F_k F_{n+2} + F_{k-1} F_{n+1}$.

$$\begin{aligned} F_{n+2+k} &= F_{n+1+k} + F_{n+k} \\ &= F_k F_{n+2} + F_{k-1} F_{n+1} + F_k F_{n+1} + F_{k-1} F_n \\ &= F_k (F_{n+1} + F_{n+2}) + F_{k-1}(F_{n+1} + F_n) \\ &= F_k F_{n+3} + F_{k-1} F_{n+2} \end{aligned}$$

Pelo princípio de indução, a igualdade está provada.

7. Prove by induction in n that $\sum_{m=0}^{n} \binom{n}{m} = 2^n$

Resposta: Inicialmente, considere a identidade de Pascal: $\binom{n+1}{k} = \binom{n}{k-1} + \binom{n}{k}$. Para a prova, seja o caso inicial: $n = 0 \Rightarrow \sum_{m=0}^{0} \binom{0}{m} = \binom{0}{0} = 1 = 2^0$, logo é verdade.

Agora suponha que para $n = k \Rightarrow \sum_{m=0}^{k} \binom{k}{m} = 2^k$, hipótese de indução.

$$\begin{aligned}
\sum_{m=0}^{k+1} \binom{k+1}{m} &= \sum_{m=0}^{k} \binom{k+1}{m} + \binom{k+1}{k+1} \\
&= \sum_{m=1}^{k} \binom{k+1}{m} + \binom{k+1}{0} + \binom{k+1}{k+1} \\
&= \sum_{m=1}^{k} [\binom{k}{m-1} + \binom{k}{m}] + \binom{k}{0} + \binom{k}{k} \\
&= \sum_{m=1}^{k} \binom{k}{m-1} + \sum_{m=1}^{k} \binom{k}{m} + \binom{k}{0} + \binom{k}{k} \\
&= \sum_{m=1}^{k} \binom{k}{m-1} + \binom{k}{k} + \sum_{m=1}^{k} \binom{k}{m} + \binom{k}{0} \\
&= \sum_{m=1}^{k+1} \binom{k}{m-1} + \sum_{m=0}^{k} \binom{k}{m} \\
&= \sum_{p=0}^{k} \binom{k}{p} + \sum_{m=1}^{k} \binom{k}{m} \qquad [m-1 = p \textit{ na primeira parcela da soma}] \\
&= 2^k + 2^k = 2^{k+1}
\end{aligned}$$

Desta maneira, $P(k) \Rightarrow P(k+1)$ e, pelo Princípio de Indução, $P(n)$ é propriedade válida.

8. Prove by induction that a graph with n vertices can have at most $\frac{n(n-1)}{2}$ edges.

   Resposta: Seja $P(n)$: um grafo com n vértices pode ter no máximo $\frac{n(n-1)}{2}$ arestas. Caso inicial: $P(2)$: um grafo com 2 vétices pode ter no máximo 1 aresta e $\frac{2\cdot(2-1)}{2} = 1$, logo, o caso base é válido. Suponha agora $P(k)$ válida. Quer-se provar que $P(k) \Rightarrow P(k+1)$.

   $P(k+1)$: Tome k vértices. O máximo de arestas, por hipótese, é $\frac{k(k-1)}{2}$ entre eles. Agora, cada um desses vértices pode fazer uma aresta nova com o vértice $k+1$. Assim, tem-se que um grafo de $k+1$ vértices tem $\frac{k(k-1)}{2} + k = \frac{k(k-1+2)}{2} = \frac{(k+1)k}{2}$ vértices. Isso mostra que $P(k) \Rightarrow P(k+1)$ e, pelo Princípio de Indução, $P(n)$ é válida.

9. Prove by induction that a complete binary tree[1] with n has $2^n - 1$ vertices.

   Resposta: Na verdade, uma árvore binária completa tem o máximo $2^n - 1$ vértices e isso só ocorre se ela cheia. Veja, por exemplo, uma árvore de 3 níveis, em que o terceiro nível tem apenas um vértice. Teremos então $1 + 2 + 1 = 4 \neq 2^3 - 1 = 7$. Assim, provarei que uma árvore binária completa de n

[1] http://web.cecs.pdx.edu/ sheard/course/Cs163/Doc/FullvsComplete.html

níveis tem no mínimo 2^{n-1} vértices e nó máximo $2^n - 1$ vértices.

Lema: Seja $V(n)$: em uma árvore binária cheia, o n-ésimo nível tem 2^{n-1} vértices. $V(1)$ é válida. Suponha válida $V(k)$. Assim, cada um de seus vértices divide-se em dois. Como existem 2^{k-1} vértices no nível k, há $2 \cdot 2^{k-1} = 2^k$ vértices, portanto está provado o lema.

Lema 2: Uma árvore binária cheia de n níveis tem $2^n - 1$ vértices. Como no nível i, há 2^{i-1} vértices, $\forall i, 1 \leq i \leq n$, se v é o número de vértices, $v = \sum_{i=1}^{n} 2^{i-1} = 2^i - 1$. Se $n = 1$, $v = \sum_{i=1}^{1} 2^{i-1} = 2^0 = 2^1 - 1 = 1$, verdade. Suponha que a soma é válida para $n = k$. Tem-se que $\sum_{i=1}^{n+1} 2^{i-1} = \sum_{i=1}^{k+1} 2^{i-1} + 2^k = 2^k - 1 + 2^k = 2^{k+1} - 1$, logo o lema 2 está provado.

Chamamos a propriedade a ser provada de $P(n)$. $P(1)$ é obviamente verdadeira. $P(2)$ também é verdadeira, pois teremos 2 ou 3 vértices em uma árvore completa de 2 níveis.

Suponha válida $P(k)$, isto é, se v é o número de vértices, $2^{k-1} \leq v \leq 2^k - 1$. Em uma árvore completa de $k+1$ níveis, a árvore de k níveis é uma árvore cheia. No $k+1$-ésimo nível, há pelo menos um vértice e no máximo 2^k. Assim, $2^k - 1 + 1 \leq v \leq 2^k - 1 + 2^k \Rightarrow 2^k \leq v \leq 2^{k+1} - 1$ vértices. $P(k+1)$ está verificada e, pelo Princípio de Indução, $P(n)$ é válida.

10. A polygon is convex if each pair of points in the polygon can be joined by a straight line that does not leave the polygon. Prove by induction in $n > 3$ that the sum of the angles of a polygon of n vertices is $180(n-2)$.

    Resposta: Inicialmente, vou usar o fato já conhecido que um polígono convexo de três vértices, o triângulo, tem 180^o. Também note que se dividirmos um polígono convexo em dois por uma reta, teremos dois polígonos convexos. Partindo disso, seja $P(n)$: um polígono convexo de n vértices tem a soma dos ângulos $180(n-2)$.

    $P(4)$: em um polígono convexo de 4 vértices, tracemos um segmento entre dois vértices opostos. Assim, teremos dois triângulos. A soma dos ângulos internos é então $2 \cdot 180 = 180 \cdot (4-2) = 360$, logo a propriedade é válida.

    Suponha $P(k)$ válida. Em um polígono convexo de $k+1$ vértices, tomamos três vértices vizinhos e formamos um triângulo. Assim, teremos dois polígonos, um com k vértices e outro com 3 vértices. Logo, a soma dos ângulos internos é $180(k-2) + 180 = 180(k-2+1) = 180((k+1)-2)$, como queríamos provar. Conclui-se que $P(k) \Rightarrow P(k+1)$ e, então, $P(n)$ é verdadeira.

2 Insertion Sort vs Mergesort

Implementou-se o Insert Sort e o Merge Sort. A função Merge Sort possui uma função auxiliar chamada "merge" para combinar duas sublistas ordenadas em uma única ordenada. Também criou-se uma função para conferenência de erro chamada "is_sorted".

1 Random Order

Para se criar uma lista randomicamente ordenada, utilizou-se a função "np.random.shuffle". Observou-se que o Insertion Sort tomou um tempo muito maior, mais de 1000 vezes maior, comparado ao Merge Sort. Isso ocorre devido à sua complexidade, n^2, enquanto do Merge é $n \log n$. O fator negativo do Merge Sort é sua recursividade, que exige memória computacional, mas para listas de floats pequenos, não se teve problema. O resultado é visto na imagem abaixo.

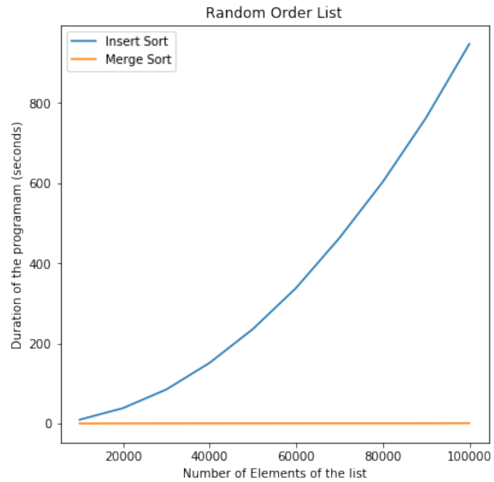
2 Ascending Order

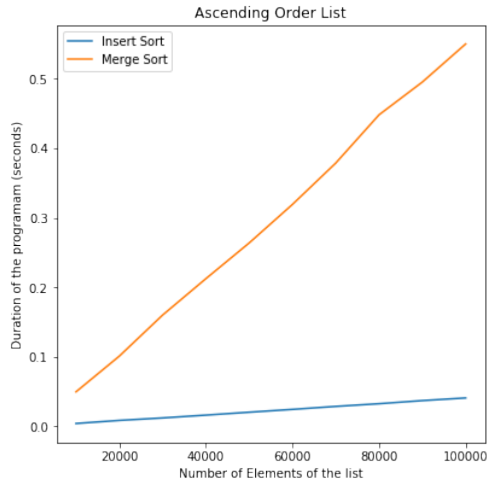
Criar um array em ordem ascendente é muito simples. O seu resultado é bem interessante. O Merge Sort teve maior tempo do que o Insertion Sort, visto que o processo de divisão é o mesmo, a diferença ao random order é na hora de comparar as listas, pois compara-se apenas um elemento de uma sublista com toda a outra e insere-se a sublista com maiores valores. Mas o Insertion Sort toma tempo linear, visto que não entra no "while" da função. O resultado é mostrado abaixo.

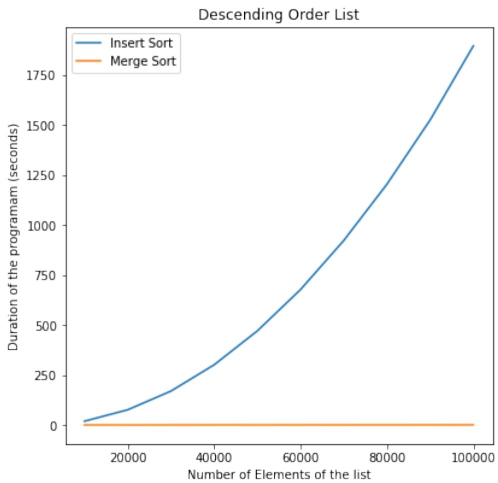
3 Descending Order

Esse é o pior caso para o Insertion Sort, pois ele se utiliza do "while" da função até o iterador da sublista ficar negativo. Isso leva à complexidade n^2, mas o tempo é maior do que o Random Order, chegando a ser quase 2 vezes mais lento. O Merge Sort realiza um trabalho similar ao Ascending Order e é extremamente mais eficiente do que o Insertion. No gráfico ele aparenta ser até constante, dada as diferenças. Mas a complexidade é $n \log n$.

4 Graphics


Random Order List
Insert Sort
Merge Sort
800
600
400
200
0
Duration of the programam (seconds)
20000
40000
60000
80000
100000
Number of Elements of the list


Ascending Order List
Insert Sort
Merge Sort
0.5
0.4
0.3
0.2
0.1
0.0
Duration of the programam (seconds)
20000
40000
60000
80000
100000
Number of Elements of the list


Descending Order List
Insert Sort
Merge Sort
1750
1500
1250
1000
750
500
250
0
Duration of the programam (seconds)
20000
40000
60000
80000
100000
Number of Elements of the list