Example- Compute the inverse DFT of the DFT of signal $x[n]$, X_k, from prior example.

$$X_k = \begin{cases} 6 & k=0 \\ -2+j2 & k=1 \\ -2 & k=2 \\ -2-j2 & k=3 \end{cases} \qquad N=4$$

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X_k e^{j\frac{2\pi kn}{N}} = \frac{1}{4}\sum_{k=0}^{3} X_k e^{j\frac{2\pi kn}{4}}$$

$n=0$ $\quad x[0] = \frac{1}{4}\left[6e^{0} + (-2+j2)e^{0} + (-2)e^{0} + (-2-j2)e^{0}\right]$

$$x[0] = 0$$

$n=1$ $\quad x[1] = \frac{1}{4}\left[6e^{0} + (-2+j2)e^{j\frac{2\pi(1)1}{4}} - 2e^{j\frac{2\pi(2)1}{4}} + (-2-j2)e^{j\frac{2\pi(3)1}{4}}\right]$

$$x[1] = 1$$

$n=2$ $\quad x[2] = \frac{1}{4}\left[6e^{0} + (-2+j2)e^{j\frac{2\pi(1)2}{4}} - 2e^{j\frac{2\pi(2)2}{4}} + (-2-j2)e^{j\frac{2\pi(3)2}{4}}\right]$

$$x[2] = 2$$

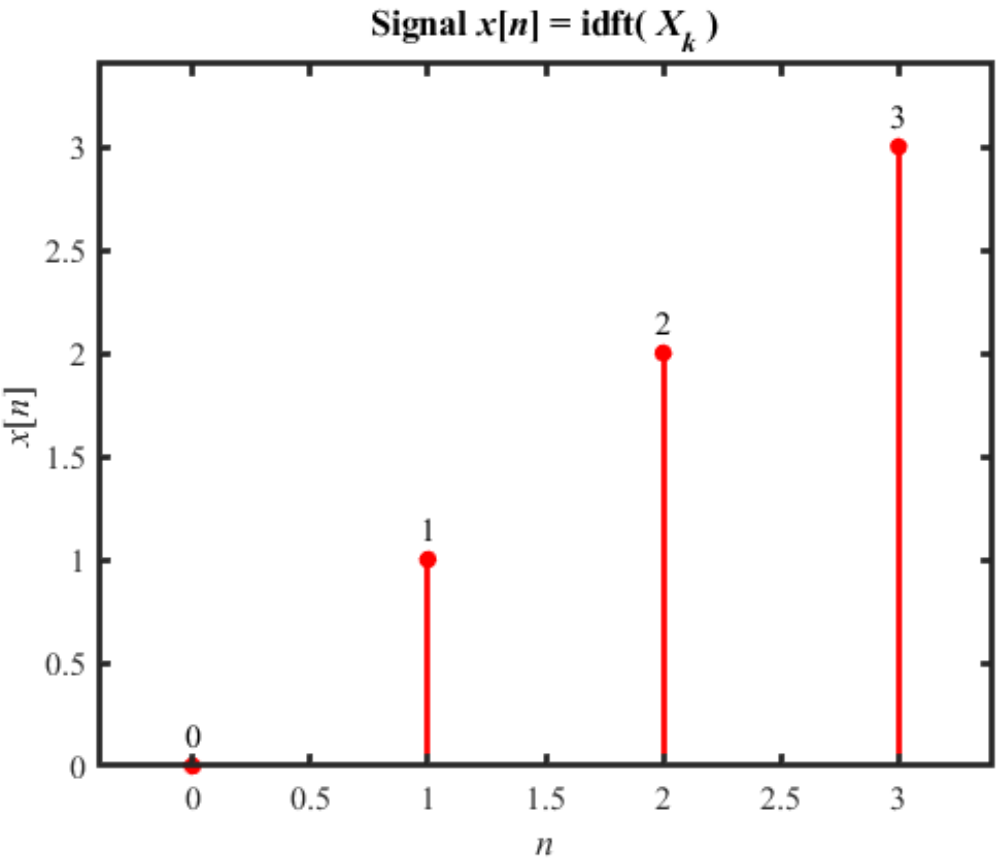
$n=3$ $\quad x[3] = \frac{1}{4}\left[6e^{0} + (-2+j2)e^{j\frac{2\pi(1)3}{4}} - 2e^{j\frac{2\pi(2)3}{4}} + (-2-j2)e^{j\frac{2\pi(3)3}{4}}\right]$

$$x[3] = 3$$

$\Rightarrow$ Easier way is to use MATLAB

```
% Chapter 4 IDFT Example 1 (chap_04_idft_example_1.m)
% Compute IDFT of Xk from previous example
clear; clc; close all;
Xk=[6,-2+j*2,-2,-2-j*2];  % Define the input vector
x = idft(Xk);             % Call IDFT function
% get rid of tiny imaginary part (round-off error)
x = real(x);
n = 0:1:length(x)-1; % Time indices
% Plot signal
stem(n,x,'r.','linewidth',2,'markersize',22),
axis([-0.4 3.4 0 3.4]),
for m=1:length(x),
    text(n(m),x(m)+0.05,[num2str(x(m),4)],...
        'horizontalalignment','center','verticalalignment','bottom')
end
ylabel('{\itx}[{\itn}]','fontsize',16,'fontname','times'),
xlabel('{\itn}','fontsize',16,'fontname','times'),
title('Signal {\itx}[{\itn}] = idft( {\itX}_{\itk} )',...
    'fontsize',16,'fontname','times'),
set(findobj('type','line'),'linewidth',1.5,'markersize',18)
set(findobj('type','axes'),'linewidth',2,'fontsize',12,'fontname','times')
set(findobj('type','text'),'fontsize',12,'fontname','times')

**********************************************************************
%  Inverse Discrete Fourier Transform
%
function x = idft(Xk)
[N,M] = size(Xk);
if M ~=1,   % makes sure that Xk is a column vector
  Xk = Xk.';
  N = M;
end
x=zeros(N,1);
k = 0:N-1;
for n=0:N-1
  x(n+1) = exp(j*2*pi*k*n/N)*Xk;
end
x = x/N;
```



Same as original input signal!