how U doin'?

LaTeX & LyX Test

Mark Taylor

May 2020

Contents

1 Insert picture

Before inserting pictures, add some shit.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Figure 1: The Beatles

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

2 Insert svg

3 Links

Here is a link to fig 3. And here is another link that links to My GitHub. See, there's a footnote.[1] Besides, there's graph, click here to view.

The first and most common use of templates is to supportgeneric programming, that is, pro-gramming focused on the design, implementation, and use of general algorithms. Here, "general"means that an algorithm can be

[1] Hola, this is my GitHub `https://github.com/How-u-doing`

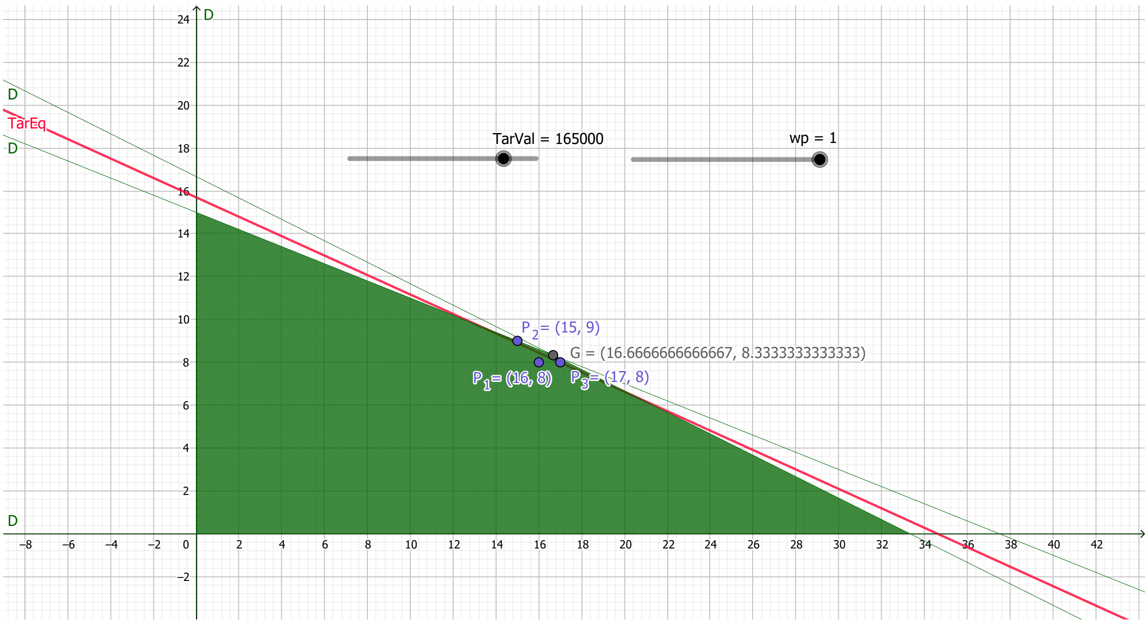

Figure 2: *MATLAB graph 1*

designed to accept a wide variety of types as long as they meet the algorithm's requirements on its arguments. The template is C++'s main support for generic pro-gramming. Templates provide (compile-time) parametric polymorphism.There are many definitions of "generic programming." Thus, the term can be confusing. How-ev er, in the context of C++, "generic programming" implies an emphasis on the design of generalalgorithms implemented using templates.Focusing more on generative techniques (seeing templates as type and function generators) andrelying on type functions to express compile-time computation are calledtemplate metaprogram-ming, which is the subject of Chapter 28.The first and most common use of templates is to supportgeneric programming, that is, pro-gramming focused on the design, implementation, and use of general algorithms. Here, "general"means that an algorithm can be designed to accept a wide variety of types as long as they meet thealgorithm's requirements on its arguments. The template is C++'s main support for generic pro-gramming. Templates provide (compile-time) parametric polymorphism.There are many definitions of "generic programming." Thus, the term can be confusing. How-ev er, in the context of C++, "generic programming" implies an emphasis on the design of generalalgorithms implemented using templates.Focusing more on generative techniques (seeing templates as type and function generators) andrelying on type functions to express compile-time computation are calledtemplate metaprogram-ming, which is the subject of Chapter 28.The first and most common use of templates is to supportgeneric programming, that is, pro-gramming focused on the design, implementation, and use of general algorithms. Here, "general"means that an algorithm can be designed to accept a wide variety of types as long as they meet thealgorithm's requirements on its arguments. The template is C++'s main support for generic pro-gramming. Templates provide (compile-time) parametric polymorphism.There are many definitions of "generic programming." Thus, the term can be confusing. How-ev er, in the context of C++, "generic programming" implies an emphasis on the design of generalalgorithms implemented using templates.Focusing more on generative techniques (seeing templates as type and function generators) andrelying on type functions to express compile-time computation are calledtemplate metaprogram-ming, which is the subject of Chapter 28.Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

4 Insert graph & code

4.1 Function templates

ensurehelveticaisembedded_()Chapter 1Function TemplatesThis chapter introduces function templates. Function templates are functions that are parameterizedso that they represent a family of functions.1.1 A First Look at Function TemplatesFunction templates provide a functional behavior that can be called for different types. In otherwords, a function template represents a family of functions. The representation looks a lot like anordinary function, except that some elements of the function are left undetermined: These elementsare parameterized. To illustrate, let's look at a simple example.nsurehelveticaisembedded_()Chapter 1Function TemplatesThis chapter introduces function templates. Function templates are functions that are parameterized so that they represent a family of functions.1.1 A First Look at Function TemplatesFunction templates provide a functional behavior that can be called for different types. In other words, a function template represents a family of functions. The representation looks a lot like anordinary function, except that some elements of the function are left undetermined: These elementsare parameterized. To illustrate, let's look at a simple example.nsurehelveticaisembedded_()Chapter 1Function TemplatesThis chapter introduces function templates. Function templates are functions that are parameterizedso that they represent a family of functions.1.1 A First Look at Function TemplatesFunction templates provide a functional behavior that can be called for different types. In otherwords, a function template represents a family of functions. The representation looks a lot like anordinary function, except that some elements of the function are left undetermined: These elementsare parameterized. To illustrate, let's look at a simple example.nsurehelveticaisembedded_()Chapter 1Function TemplatesThis chapter introduces function templates. Function templates are functions that are parameterizedso that they represent a family of functions.1.1 A First Look at Function TemplatesFunction templates provide a functional behavior that can be called for different types. In otherwords, a function template represents a family of functions. The representation looks a lot like anordinary function, except that some elements of the function are left undetermined: These elementsare parameterized. To illustrate, let's look at a simple example.nsurehelveticaisembedded_()Chapter 1Function TemplatesThis chapter introduces function templates. Function templates are functions that are parameterizedso that they represent a family of functions.1.1 A First Look at Function TemplatesFunction templates provide a functional behavior that can be called for different types. In otherwords, a function template represents a family of functions. The representation looks a lot like anordinary function, except that some elements of the function are left undetermined: These elementsare parameterized. To illustrate, let's look at a simple example.nsurehelveticaisembedded_()Chapter 1Function TemplatesThis chapter introduces function templates. Function templates are functions that are parameterizedso that they represent a family of functions.1.1 A First Look at Function TemplatesFunction templates provide a functional behavior that can be called for different types. In otherwords, a function template represents a family of functions. The representation looks a lot like anordinary function, except that some elements of the function are left undetermined: These elementsare parameterized. To illustrate, let's look at a simple example.Let's look at The Beatles, or u can click here to see Figure 1.

Go back to Graph 1 or cover page

4.2 Class template

When we call a function template such asmax()for some arguments, the template parameters aredetermined by the arguments we pass. If we pass twoints to the parameter typesT, the C++ compilerhas to conclude thatTmust beint.However,Tmight only be "part" of the type. For example, if we declaremax()to use constantreferences:template<typenameT>T max (Tconst& a, Tconst& b){returnb < a ? a : b;}and passint, againTis deduced asint, because the function parameters match for int const&.

4.3 Code

Following Python code snippet is inserted directly via LaTeXcode in LyX. However, we can also import them by source files, see 4.3.1 - 4.3.3. Wanna see *minted* wc program?

```
import numpy as np

def incmatrix(genl1,genl2):
  m = len(genl1)
  n = len(genl2)
  M = None #to become the incidence matrix
  VT = np.zeros((n*m,1), int)  #dummy variable

#compute the bitwise xor matrix
```

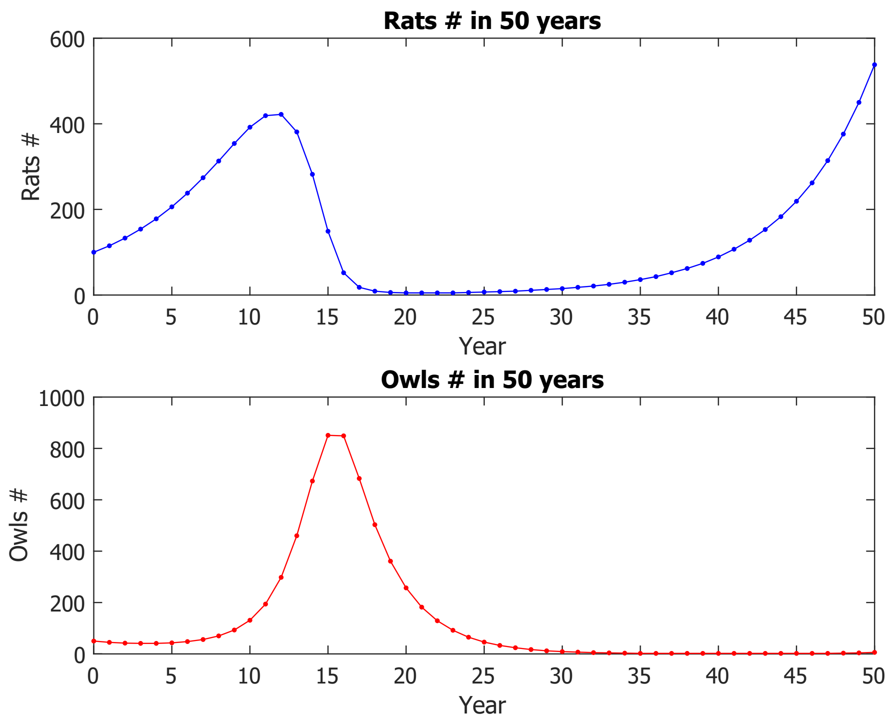

Figure 3: *MATLAB graph 2*

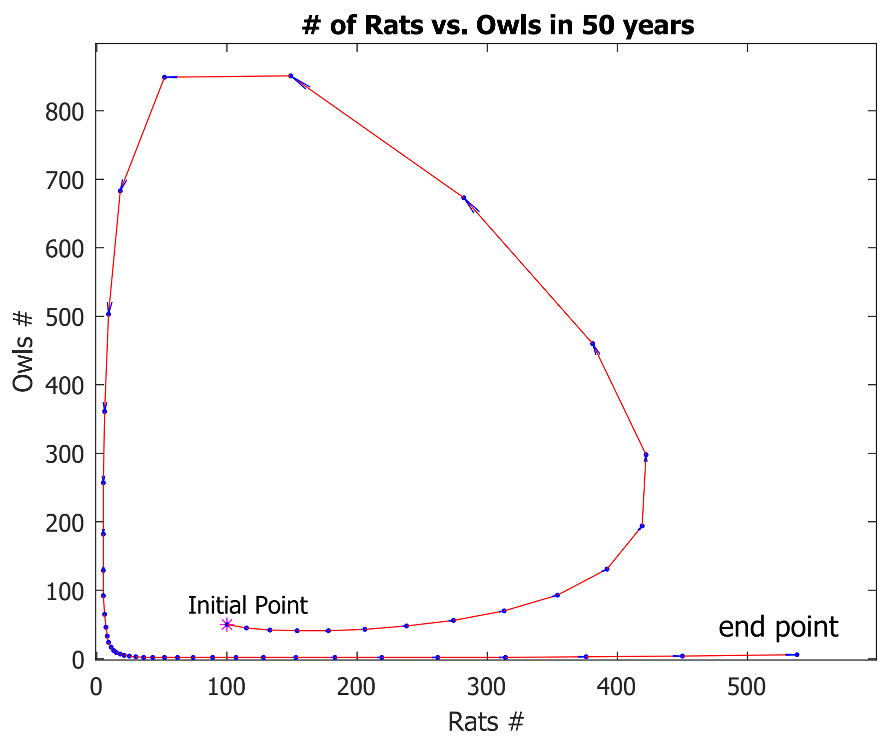

Figure 4: *MATLAB graph 3*

```python
  M1 = bitxormatrix(genl1)
  M2 = np.triu(bitxormatrix(genl2),1)

  for i in range(m-1):
    for j in range(i+1, m):
      [r,c] = np.where(M2 == M1[i,j])
  for k in range(len(r)):
    VT[(i)*n + r[k]] = 1;
    VT[(i)*n + c[k]] = 1;
    VT[(j)*n + r[k]] = 1;
    VT[(j)*n + c[k]] = 1;

  if M is None:
    M = np.copy(VT)
  else:
    M = np.concatenate((M, VT), 1)

  VT = np.zeros((n*m,1), int)

  return M

```

Listing 1: Python example

4.3.1 MATLAB Code

```matlab
% programs to solve exercise 8 in page 231
years=50;
time=0:50;
x=zeros(1,51); % # of rats in 50 years
y=zeros(1,51); % # of owls in 50 years
% question 1
r1=0.2; r2=0.3;
a1=0.001; a2=0.002;
x(1)=100; y(1)=50; % initial conditions
figure(2)
plot(x(1),y(1),'m*')
text(x(1)-30,y(1)+30,'Initial Point')
for k=1:years
    x(k+1)=(1+r1-a1*y(k))*x(k);
    x(k+1)=round(x(k+1)); % # can only be integer, e.g. 4.3->4, 4.6->5
    y(k+1)=(1-r2+a2*x(k))*y(k);
    y(k+1)=round(y(k+1));

    hold on
    % draw vecters (with arrow) every two points
    plot(x(k),y(k),'b.', x(k+1),y(k+1),'b.')
    vectarrow([x(k),y(k)],[x(k+1),y(k+1)])
end
xlabel('Rats #'), ylabel('Owls #')
title('# of Rats vs. Owls in 50 years')

figure(1)
subplot(2,1,1)
plot(time,x,'-b.')
xlabel('Year'), ylabel('Rats #')
title('Rats # in 50 years')

subplot(2,1,2)
plot(time,y,'-r.')
xlabel('Year'), ylabel('Owls #')
title('Owls # in 50 years')

function vectarrow(p0,p1)
% see also <https://www.mathworks.com/matlabcentral/fileexchange/
%         7470-plot-2d-3d-vector-with-arrow?s_tid=prof_contriblnk>
x0 = p0(1);
y0 = p0(2);
x1 = p1(1);
y1 = p1(2);
plot([x0;x1],[y0;y1],'r');   % Draw a line between p0 and p1

```

```
p = p1-p0;
alpha = 0.1;  % Size of arrow head relative to the length of the vector
beta = 0.1;   % Width of the base of the arrow head relative to the length

hu = [x1-alpha*(p(1)+beta*(p(2)+eps)); x1; x1-alpha*(p(1)-beta*(p(2)+eps))];
hv = [y1-alpha*(p(2)-beta*(p(1)+eps)); y1; y1-alpha*(p(2)+beta*(p(1)+eps))];

hold on
plot(hu(:),hv(:),'b')  % Plot arrow head
hold off
end
```

Listing 2: Modeling hw4 MATLAB code

4.3.2 C Code

1. by *lstlisting*

```
/* Word counting program, by K&R C */
/* bare-bones version of UNIX program 'wc' */
#include <stdio.h>

#define IN  1    /* inside a word */
#define OUT 0    /* outside a word */

/* count lines, words, and characters in input */
int main(int argc, char* argv[])
{
  int c, nl, nw, nc, state;
  FILE* fp;

  // complile: gcc file_wc.c -o file_wc
  // Linux command line: ./file_wc input.txt
  if (!(fp=fopen(argv[1], "r"))) {
    perror("Error opening file!\n");
  }

  state = OUT;
  nl = nw = nc = 0;
  while ((c = fgetc(fp)) != EOF) {
    ++nc;
    if (c == '\n')
      ++nl;
    if (c == ' ' || c == '\n' || c == '\t')
      state = OUT;
    else if (state == OUT) {
      state = IN;
      ++nw;
    }
  }
  printf("%d %d %d\n", nl, nw, nc);
}
```

Listing 3: File word_counting program

2. by *minted*

This however is **imported** by *minted*

```c
/* Word counting program, by K&R C */
/* bare-bones version of UNIX program 'wc' */
#include <stdio.h>

#define IN  1    /* inside a word */
#define OUT 0    /* outside a word */

/* count lines, words, and characters in input */
int main(int argc, char* argv[])
{
	int c, nl, nw, nc, state;
	FILE* fp;

	// complile: gcc file_wc.c -o file_wc
	// Linux command line: ./file_wc input.txt
	if (!(fp=fopen(argv[1], "r"))) {
		perror("Error opening file!\n");
	}

	state = OUT;
	nl = nw = nc = 0;
	while ((c = fgetc(fp)) != EOF) {
		++nc;
		if (c == '\n')
			++nl;
		if (c == ' ' || c == '\n' || c == '\t')
			state = OUT;
		else if (state == OUT) {
			state = IN;
			++nw;
		}
	}
	printf("%d %d %d\n", nl, nw, nc);
}
```

Listing 1: *minted* word_counting program

4.3.3 C++ Code

```cpp
template<typename RandomIt, typename Compare>
void sortingMethods<RandomIt, Compare>::BubbleSort(RandomIt first, RandomIt last, Compare comp)
{
#if defined Cocktail_shaker_sort
  // see also <https://en.wikipedia.org/wiki/Cocktail_shaker_sort>
  // Example: list (2,3,4,5,1), which would only need to go through one pass (indeed 1.5 pass, one
  // more left-to-rigth comparison) of cocktail sort to become sorted, but if using an ascending
  // bubble sort would take four passes. However one cocktail sort pass should be counted as two
  // bubble sort passes. Typically cocktail sort is less than two times faster than bubble sort.
  int n = last - first;
  int m = 1;
  int lastLeftSwappedIndex; // index of last left-side sorted
  int lastRightSwappedIndex;  // index of first right-side sorted
  while (n > m) {
    lastLeftSwappedIndex = n - 1;
```

```
    lastRightSwappedIndex = 0;
    for (int i = m; i < n; ++i) {
      if (comp(*(first + i), *(first + i - 1))) {
        swap(first + i, first + i - 1);
        lastRightSwappedIndex = i;
      }
    }
    n = lastRightSwappedIndex;

    if (n == 0) // no swap, no need to compare right-to-left back
      return;

    for (int j = n - 1; j >= m; --j) {
      if (comp(*(first + j), *(first + j - 1))) {
        swap(first + j, first + j - 1);
        lastLeftSwappedIndex = j;
      }
    }
    m = lastLeftSwappedIndex;
  }

#else // just left-to-right bubble sort
  // see also <https://en.wikipedia.org/wiki/Bubble_sort#Optimizing_bubble_sort>
  int n = last - first; // unsorted length
  int lastSwappedIndex = 0;
  while (n > 1) {
    lastSwappedIndex = 0;
    for (int i = 1; i < n; ++i) {
      if (comp(*(first + i), *(first + i - 1))) {
        swap(first + i, first + i - 1);
        lastSwappedIndex = i;
      }
    }
    n = lastSwappedIndex;
  }
#endif // defined Cocktail_shaker_sort
}
```

Listing 4: Bubble sort program

5 Matrix

1	2	0	0	0
0	2	0	0	8
0	0	3	0	0
0	0	0	4	0
0	0	0	0	5