Tips and example design document
IC 150 P Computation for engineers Lab

What is the purpose of the design document ? Suppose that we pick at random another group that has spent no effort on your project topic. If your design document has been prepared well, then this other team can digest your design document, and pretty much set to work on implementing and testing your solution.

The recommended sections to your design document

The computational task attempted

containing IN YOUR OWN WORDS, a description in crisp, clear, plain English. You should describe the problem "WITHOUT putting in any bits of your solution."

Our solution

containing an unambiguous description of the algorithm, a list of exceptions your solution can handles, and some light indication of why your solution is expected to work. You may describe your algorithm using a Flow chart, or some lines of Pseudocode. Avoid use of too much detail. If using a flow chart, do not use more than 15 or 20 boxes, and if writing pseudocode, do not use more than 20 lines. **You may neatly write/draw the pseudocode/flow-chart by hand, take a good quality scan of the picture, and use it in your design document.**

Components

containing formal specifications of the various functions you will write, perhaps an optional list of only the key data structures used.

Responsibilities and timeline

a mapping between individual functions (including `main` and `readme.txt` which is not a function per se) and individual members of the group. A listing of a few key milestones (less than 6), and some estimate/aspiration of when these milestones will be reached. A Gantt chart will be nice, but is not expected.

Further tips on writing the design document

Aim to write a document that is no more than 2 typed pages long, preferably shorter. There is no penalty for a short report, as long as it contains the essential information. Please do not use any font bigger than 12 pts, or smaller than 10 pts in any part of your typed document. And avoid excessive blank spaces inside your typed document.

If during the course of the project, you discover that you need to change/modify your algorithm, data structure etc. then make notes in your workbook. In the final week of the project, it will be nice if you can append a section (5) corrections/lessons learnt: containing you know what But do not prepare your design document worrying now about how to minimize the 'lessons learnt' section !

Design document for project on drawing the convex hull
by Al Kharazmi, L. Euler, B. Pascal, C. Babbage, and A. Turing
10th November 2016

1 The computational problem

1.1 Problem statement:

The user specifies a number of points on the plane. The program must compute, describe (and possibly draw) the convex hull of these points. (See appendix for definitions).

2 Our solution

We implement the algorithm:

Algorithm 1: The gift wrapping algorithm. Source: *Algorithms in C,* R. Sedgewick

Data: unsigned integer variables N, i, j, flag
struct point {
float xPart;
float yPart;
};
N entry array of the type structure point: $p[]$
Result: $cp[]$ an array of up to N entries of the structure point
ASCII drawing of convex hull on screen

```
1  cp[0] := input point with the least value of xPart ;
2  flag := 0;
3  i := 0 ;
4  while flag = 0 do
5  |  for j = 0 to N − 1 do
6  |  |  if p[j] ≠ cp[i] AND every other point is to the right of the line through cp[i], p[j] then
7  |  |  |  i := i + 1 ;
8  |  |  |  cp[i] := p[j] ;
9  |  |  |  if cp[i] = cp[0] then
10 |  |  |  |  flag=1
11 |  |  |  end
12 |  |  end
13 |  end
14 |  Draw the filled polygon formed by the points cp[0], ..., cp[i].
15 end
```

Components

f1 `void input()`

f2 `struct point calculateStartingCP( int N, struct point pt[N])`

f3 `struct straightLine calculateStrainghtLineSlopeAndIntercept ( struct point ptOne, struct point ptTwo)`

f4 `int isPointToRightOfLine( struct point pt, struct StraightLine stLine)`

f5 `void runGiftWrapping( struct point* pointerToCP, struct point* pointerToInputPoints, )`

f6 `void main()`

f7 `readme.txt`

3 Responsibilities and timeline

Responsibilities: Al Kharazmi: **f3** L. Euler: **f4** B. Pascal: **f5** C. Babbage: **f6, f7** A. Turing: **f1, f2**

Timeline: Milestone 1: **f1, f2** to be ready by 20/11/2016. Milestone 2: **f3, f4** to be ready by 25/11/2016. Milestone 3: **f5, f6** to be ready by 28/11/2016. Milestone 4: **f7** to be ready by 31/11/2016.

Appendix: What is the convex hull of a bunch of points ?

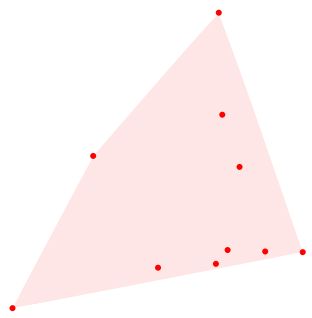

Figure 1: *an example*

Suppose that we are given the bunch of N points on the plane:

$$(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N).$$

The convex hull of this set of points is defied as the smallest set that includes these points, and the following points generated by *convex combinations:*

$$\left(\sum_{i=1}^{N} \lambda_i x_i, \sum_{i=1}^{N} \lambda_i y_i \right),$$

where we allow all the different possible values of the coefficients λ_i but subject to the two constraints:

$$\lambda_i \geq 0$$
$$\sum_{i=1}^{N} \lambda_i = 1.$$

Note that the convex combination of two points is exactly the line segment between them. The convex hull of three points is the triangle (including all interior points) formed by the three points.