Homework 5

CS 5114 (Spring 2013)

Assigned on April 2, 2013.
Submit PDF solutions by email to the TA
by the beginning of class on Tuesday, April 9, 2013.

Instructions:

- You are not allowed to consult any sources other than your textbook, the slides on the course web page, your own class notes, the TA, and the instructor. In particular, do not use a search engine.
- Do not forget to typeset your solutions. *Every mathematical expression must be typeset as a mathematical expression, e.g., the square of n must appear as n^2 and not as "n^2".* Students can use the LaTeX version of the homework problems to start entering their solutions.
- Describe your algorithms as clearly as possible. The style used in the book is fine, as long as your description is not ambiguous. Explain your algorithm in words. A step-wise description is fine. *However, if you submit detailed pseudo-code without an explanation, we will not grade your solutions.*
- Do not make any assumptions not stated in the problem. If you do make any assumptions, state them clearly, and explain why the assumption does not decrease the generality of your solution.
- Do not describe your algorithms only for a specific example you may have worked out.
- You must also provide a clear proof that your solution is correct (or a counter-example, where applicable). Type out all the statements you need to complete your proof. *You must convince us that you can write out the complete proof. You will lose points if you work out some details of the proof in your head but do not type them out in your solution.*
- Describe an analysis of your algorithm and state and prove the running time. You will only get partial credit if your analysis is not tight, i.e., if the bound you prove for your algorithm is not the best upper bound possible.

Problem 1 (20 points) Solve exercise 1 in Chapter 6 (pages 312-313) of your textbook.

Problem 2 (30 points) Solve exercise 17 in Chapter 6 (pages 327-328) of your textbook. For part (b) of this exercise, keep in mind that a rising trend must begin on the first day. You should find this requirement important in defining your sub-problems.

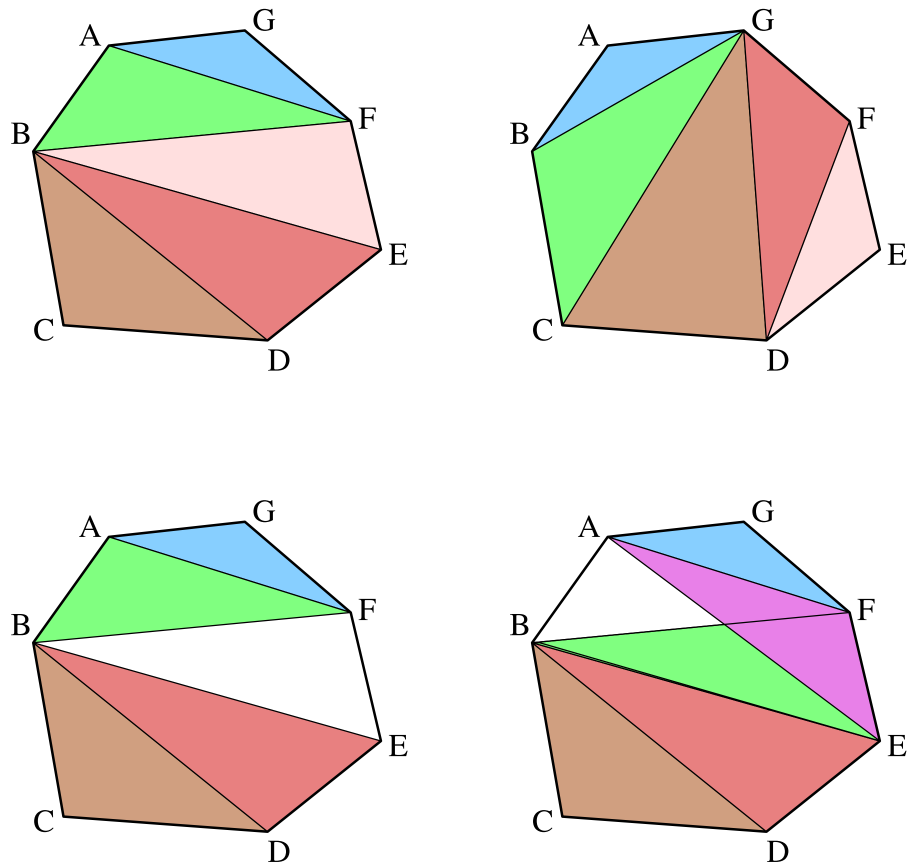
Problem 3 (50 points) A convex polygon is a polygon where very interior angle is less than 180 degrees. A museum is in the shape of a convex polygon with n vertices. The museum is patrolled by guards. The Directory of Security at the museum has the following rules to ensure the most safety in as time-economical a way as possible:

(a) Each guard traverses a path in the shape of a triangle; each vertex of such a triangle must a vertex of the polygon.

(b) A guard can survey all the points inside his or her triangle and *only* these points; we say that these points are *covered* by the guard.

(c) Every point inside the museum must be covered by some guard.

(d) The triangles traversed by any pair of guards do not overlap in their interiors, although they may share a common edge.



Each guard can be specified by the triangle he/she patrols. We call a set of triangles that satisfy these rules *legal*. Above are four figures that illustrate the problem. The museum is the polygon ABCDEFG. Each coloured (shaded) triangle corresponds to a guard and the guard will traverse the perimeter of his or her triangle.

- The top two figures show a set of triangles that are legal, since they satisfy the constraints laid down by the Directory of Security. In the top left figure, the guards traverse the boundaries of triangles AFG (blue), ABF (green), BEF (pale red), BDE (light red), and BCD (brown). In the top right figure, the guards traverse ABG (blue), BCG (green), CDG (brown), DFG (light red), and DEF (pale red).
- The bottom two figures show a set of triangles that *do not* satisfy these constraints: in the figure on the bottom left, part of the museum is not covered by any guard (the unshaded triangle BEF) while in the figure on the bottom right, the pink triangle (AEF) and the green triangle (BEF) intersect (not that the part of the green triangle in the image is covered by the pink triangle).

Given these constraints, the *cost* incurred by a guard is the length of the perimeter of the triangle the guard traverses. The *total* cost of a set of triangles is the sum of the length of their perimeters. Our goal is to find a legal set of triangles such that the total cost of the triangles is as small as possible. Given the x- and y-coordinates of the vertices of the art gallery and the ordering of these vertices along the boundary of the art gallery, devise an algorithm whose running time is polynomial in n to solve this problem. Note that we are not trying to minimise the number of guards; we want to minimise the total lengths of the routes patrolled by the guards. You may assume that the length of any line segment is the Euclidean distance between the end-points of the line segment and that this length can be computed in constant time.

Since this problem may be quite difficult, let us split up into more digestible pieces. Let the museum be a convex polygon P with n vertices $p_0, p_1, \ldots p_{n-1}$ ordered consecutively in counter-clockwise order around P. A line segment p_ip_j is a *diagonal* if p_i and p_j are not adjacent vertices, i.e., $|i - j| \neq 1$. The order of the endpoints does not matter, i.e., the diagonals p_ip_j and p_jp_i are identical. Since P is convex, every point of a diagonal (other than its end-points) lies in the interior of P. Clearly, in any legal set of triangles, every triangle edge is either an edge of P or a diagonal of P. For each of the questions below, you must provide a proof for your answer. Some of the proofs may be short, especially for parts (i) and (iii).

(i) (4 points) How many diagonals does P contain in total?

(ii) (9 points) How many diagonals does any legal set of triangles contain?

(iii) (7 points) Given a legal set of triangles, express the total cost of these triangles in terms of the perimeter of P and the lengths of those edges of the triangles that are diagonals of P.

(iv) (10 points) Let us consider one way to construct a dynamic programming solution, which will not lead to an optimal solution. Consider vertex p_0. There are two possibilities in the optimal solution:

   a. A diagonal is not incident on p_0. In this case, p_1p_{n-1} must be a diagonal in the optimal solution. (Try to prove yourself that the configuration is not legal otherwise.) Therefore, we have a single sub-problem involving the convex polygon formed by the sequence of vertices $p_1, p_2, \ldots p_{n-1}$ and the diagonal p_1p_{n-1}.

   b. A diagonal is incident on p_0. Suppose p_i is the other vertex of the diagonal. This diagonal yields two sub-problems, one involving the vertices $p_i, p_{i+1}, \ldots, p_{n-2}, p_{n-1}, p_0$ and the other involving the vertices $p_0, p_1, p_2, \ldots p_i$.

   Try to develop a set of sub-problems based on these observations and point out where the solution goes astray. You must show how you will write the dynamic programming recurrence and then tell us why it is difficult to complete the recurrence. You do not need to prove anything here; simply discuss why this idea is hard to finish.

(v) (20 points) Inspired by this wrong approach, define a new set of sub-problems that you can use to compute the optimal solution. Prove why the new formulation solves the problem correctly. State and prove the running time of the resulting algorithm.