CS 256: Algorithm Design and Analysis

Assignment 6 (due 11/05/2020)

Instructor: Sam McCauley

Problem 1. A complex number x is written $a + bi$, where a and b are real numbers.

To multiply two complex numbers $a_1 + b_1 i$ and $a_2 + b_2 i$, we can use the equation:

$$(a_1 + b_1 i)(a_2 + b_2 i) = (a_1 a_2 - b_1 b_2) + (a_2 b_1 + a_1 b_2) i$$

We can calculate this final value with four multiplications: $a_1 a_2$, $b_1 b_2$, $a_2 b_1$, and $a_1 b_2$. Give a method to calculate this value using only three multiplications.

(This is a fairly well-known problem, but I'd encourage you to try to think it through yourself. It's a fun puzzle.)

Solution. □

Problem 2. Let A and B be $n \times n$ matrices such that each entry in A or B is either 0 or 1. Let $C = AB$ be the product of A and B. Recall that each entry c_{ij} of C is defined as

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

Let's call an entry c_{ij} **simple** if there is exactly one term k in that equation such that $a_{ik}b_{kj} = 1$. That is to say, c_{ij} is simple if and only if $c_{ij} = 1$ (since A and B only have entries that are 0 or 1).

Let us call a matrix P the **simple witness** for C if $P_{ij} = 0$ if c_{ij} is not simple, and $P_{ij} = k$ if c_{ij} is simple and $a_{ik}b_{kj} = 1$.

Give an $O(n^{\log_2 7})$ algorithm to construct P, the simple witness for C, given A and B. Note that Strassen's algorithm runs in this time—so I am asking you to construct P using $O(1)$ matrix multiplications, with $O(n^2)$ additional work.

A slower method: Here's a method to construct P in $O(n^3)$ time to get you started. We can find all the simple c_{ij} in $O(n^{\log_2 7})$ time by multiplying A and B and finding which elements of C are equal to 1. We can then go through each simple c_{ij}; for each, we can find the k such that $a_{ik}b_{kj} = 1$ in $O(n)$ time. Since there are $O(n^2)$ simple c_{ij} at most, this gives $O(n^3)$ time overall.

Solution. □

Problem 3. (KT 7.5) Is the following statement true or false? If true, you must give a justification.; if false, you must give a counterexample.

Let G be an arbitrary flow network, with a source s, a sink t, and a positive integer capacity c_e on every edge e. Let (A, B) be a minimum s-t cut with respect to the capacities $\{c_e : e \in E\}$. Now suppose we add 1 to every capacity; then (A, B) is still a minimum s-t cut with respect to the new capacities $\{1 + c_e : e \in e\}$.

Solution. □

Problem 4. (Modified KT 7.23 and 7.24) Suppose youre looking at a flow network G with source s and sink t, and you want to be able to express something like the following intuitive notion: *Some nodes are clearly on the "source side" of the main bottlenecks; some nodes are clearly on the "sink side" of the main bottlenecks; and some nodes are in the middle.* However, G can have many minimum cuts, so we have to be careful in how we try making this idea precise. Here's one way to divide the nodes of G into three categories of this sort.

- We say a node v is *upstream* if, for all minimum s-t cuts (A, B), we have $v \in A$—that is, v lies on the source side of every minimum cut.
- We say a node v is *downstream* if, for all minimum s-t cuts (A, B), we have $v \in B$that is, v lies on the sink side of every minimum cut.
- We say a node v is *central* if it is neither upstream nor downstream; there is at least one minimum s-t cut (A, B) for every $v \in A$, and at least one minimum s-t cut (A', B') for which $v \in B'$.

In this question, we design an algorithm to classify vertices of G into these categories and use the classification to characterize graphs that have a unique minimum cut. Let f be the maximum flow in G. Consider the cut (A^*, B^*), where $A^* = \{u \mid u \text{ is reachable from } s \text{ in } G_f\}$ (where G_f is the corresponding residual graph) and let $B^* = V - A^*$. Thus, $v(f) = cap(A^*, B^*)$ and (A^*, B^*) is a minimum cut of G.

(a) Show that the set A^* is the set of upstream vertices of G, that is, v is upstream if and only if $v \in A^*$.

(b) Using part (a), describe an efficient algorithm to find the downstream vertices in G. (*Hint.* Consider the graph G^R, with all direction of edges in G reversed.)

(c) Show that G has a unique minimum cut if and only if G has no central vertices, that is, the union of upstream and downstream vertices is the set V.

Solution. □