CSCI-GA.3520-001: Honors Analysis of Algorithms

Final Exam, Dec 18, 2020, 9:30am-1:45pm

- This is a four hour exam (plus 15 minutes to upload solutions). There are six questions, worth 10 points each. Answer all questions and all their subparts.
- You can **only** use course notes. **No** access is allowed to textbooks, any other written or published materials, any online materials, and any other materials stored on devices.
- In the past years, to pass this exam, one needed to answer 3 or 4 problems well, instead of answering all the problems poorly.
- You must submit separate answers for separate questions.
- Read the questions carefully. Keep your answers legible, and brief but precise. Assume standard results and algorithms (i.e., those taught or referred to in class or homeworks).
- You must prove correctness of your algorithm and prove its time bound unless stated otherwise. The algorithm can be written in plain English (preferred) or as a pseudo-code.
- The instructor will be available during the exam on Zoom if you have any questions (you can login the same way that you login for the lectures).
- **Emergency only:**

  Only if you have trouble accessing or submitting on Gradescope, you can ask the instructor for the exam or submit solutions by email (khot@cs.nyu.edu or sak12@nyu.edu).

Best of luck!

Problem 1

Consider the following load balancing problem. There are n packages of weights $w_1, w_2, \ldots, w_n$. Each weight is a positive integer. Let $W = \sum_{i=1}^{n} w_i$. The goal is to distribute the packages among three trucks, T_a, T_b, and T_c, so as to minimize the maximum load (weight) on any one truck.

Give an $O(nW^2)$ time algorithm to determine the smallest possible maximum load (partial credit for a less efficient algorithm that still runs in time polynomial in n and W).

Note that if the load on T_a is L_a, and the load on T_b is L_b, then the load on T_c is $W - (L_a + L_b)$.

Problem 2

Recall that in a 2–3 tree (a) an insertion may cause a split of an internal node and (b) a deletion may cause a transfer of child(ren) among sibling nodes or may cause a merge of sibling nodes. Moreover the split or the merge operations may propagate upwards. In this problem, we are concerned about the **number** of splits (during insertion) or merges (during deletion) that propagate upwards.

1. Show a 2–3 tree T storing n items, and an arbitrarily long sequence of alternating inserts and deletes which each cause $\Theta(\log n)$ splits and merges respectively.
2. Now consider 2–6 trees. Like 2–3 trees, all leaves are at the same depth, and each internal node has between 2 and 6 children. A node is split when it gets 7 children, and a node that is reduced to having only one child either takes some children from an adjacent sibling or is merged with an adjacent sibling.

   Show that with an appropriate rule for handling merges and child transfers, any sequence of insert and delete operations applied to an initially empty tree results in $O(1)$ **amortized** splits or merges per operation.

Hint: For the first part, consider an n-node tree whose rightmost branch is $v_1, v_2, \ldots, v_k$, where v_1 is the root, each v_i, for $1 \le i \le k-1$, has three children, with its rightmost child being v_{i+1}, and finally v_k has three children w, w', w''; note that $k = \Theta(\log n)$. Consider an insert that inserts a value to the right of w''. Then consider a delete that deletes the same value and then iterate.

For the second part, since the internal nodes of degree 2 or 6 are "troublesome", try defining a potential function in terms of such nodes.

Problem 3

You are given an **acyclic directed** graph $G(V, E)$, where each node has a color, red, white, or blue. A *red sweep* is a directed path that contains **all** the red nodes (it may or may not contain additional nodes).

Design an algorithm that determines if there is a red sweep and if there is one, finds a red sweep that contains a **minimum** number of blue nodes.

Your algorithm should run in time $O(|V|+|E|)$ for full credit. Assume adjacency list representation of the graph.

Problem 4

Your city has n junctions numbered 1 through n. There are m **one-way** roads between the junctions. As a mayor of the city, you have to ensure the security of all the junctions.

To ensure the security, you have to build some police checkpoints. Each checkpoint can only be built at a junction. A checkpoint at junction i can protect junction j if either $i = j$ or the police patrol car can drive from i to j and then drive back to i.

Building checkpoints costs some money. As some areas of the city are more expensive than others, building checkpoints at some junctions might cost more money than other junctions. You have to determine the minimum budget B needed to ensure the security of all the junctions.

Design an algorithm to solve this problem. The input consists of

- a list of costs $c_1, \ldots, c_n$, where c_i is the cost of building a checkpoint at junction i, and
- for every junction i, a list of all pairs (i, j) that represent a one-way road from junction i to junction j.

The output is (the minimum budget) B.

You may use any standard algorithms as subroutines. Be sure to state the running time of your algorithm, and to justify your running time bound. For full credit, your algorithm should run in time $O(n + m)$.

Problem 5

Let $d = d(n)$ and $k = k(n)$ be functions of n such that

$$d \geq 100, \qquad k \geq 100, \qquad d^k \leq \sqrt{n}.$$

We wish to show that for all large enough n, there exists an n-node (undirected) graph G that has at least $\frac{dn}{8}$ edges and has **no** simple cycle of length k. The idea is to select a random graph G, show that the number of such cycles is likely to be small, and then delete one edge from each such cycle.

Hence, let G be a random n-node graph where every pair of nodes (u, v) is included as an edge in the graph with probability p and excluded with probability $1 - p$, independently for all pairs of nodes. Set

$$p = \frac{d}{n}.$$

Let N_e denote the number of edges in the graph G and N_c denote the number of simple cycles of length k in G. The nodes on a simple cycle are all distinct, i.e. no node appears twice. A cycle is counted only once regardless of which of the k nodes is thought of as the start and the end node.

1. What is $\mathbf{E}[N_e]$? Your answer could be exact or reasonable.
2. What is $\mathbf{E}[N_c]$? Your answer could be exact or reasonable.
3. Assume that G satisfies $N_e \geq \frac{\mathbf{E}[N_e]}{2}$ and $N_c \leq 2\ \mathbf{E}[N_c]$.

   Let G^* be obtained from G by deleting one edge arbitrarily from each simple cycle of length k. Show that G^* has at least $\frac{dn}{8}$ edges and no simple cycle of length k.
4. Now suppose that $d = k$. What is the largest value of d as a function of n so that the argument here works. For full credit, your answer should be correct up to a constant factor.

Problem 6

Let $\oplus$ denote the XOR operation on the binary values $\{0, 1\}$ (i.e. addition modulo 2). Let a, b, c be binary variables and consider the seven linear equations (modulo 2):

$$a = 1, \quad b = 1, \quad c = 1, \quad a \oplus b = 1, \quad b \oplus c = 1, \quad a \oplus c = 1, \quad a \oplus b \oplus c = 1.$$

1. How many of these equations are satisfied by the assignment $a = 0, b = 0, c = 0$?

2. Show that any assignment $a, b, c \in \{0, 1\}$ such that $(a, b, c) \neq (0, 0, 0)$ satisfies exactly t of these equations for some t. What is t?

3. Identify logical True with the value 1 and logical False with the value 0. For a variable $a \in \{0, 1\}$ interpreted as logical True or False, write its negation in terms of the $\oplus$ operation.

4. Recall that a $\mathsf{3SAT}$ instance consists of a set of logical variables and a set of clauses of the form $a \vee b \vee c$, where a, b, c are the variables or their negations. Using a reduction from $\mathsf{3SAT}$, show that the $\mathsf{3LIN}$ problem, as described below, is NP-complete.

   An instance ψ of $\mathsf{3LIN}$ consists of binary variables $z_1, \ldots, z_n$, and equations $E_1, \ldots, E_m$, and a number $1 \leq K \leq m$. Each equation E_r has one of the following types:

   $$z_i = 1, \quad z_i \oplus z_j = 1, \quad z_i \oplus z_j \oplus z_k = 1,$$

   $$z_i = 0, \quad z_i \oplus z_j = 0, \quad z_i \oplus z_j \oplus z_k = 0.$$

   The instance ψ is said to be K-satisfiable if there is a $\{0, 1\}$-assignment to the variables $z_1, \ldots, z_n$ that satisfies at least K of the equations $E_1, \ldots, E_m$. Finally,

   $$\mathsf{3LIN} = \{\langle \psi, K \rangle \mid \psi \text{ is } K - \text{satisfiable}\}.$$