

Solutions to Third Examination

CS 430 Introduction to Algorithms
Spring, 2017

Wednesday, April 26, 2017
10am–11:15am, 002 Herman Hall
11:25am–12:40pm, 104 Rettaliata Engineering Center

Exam Statistics

121 students took the exam. The range of scores was 11–97, with a mean of 63.15, a median of 67, and a standard deviation of 18.64. Very roughly speaking, if I had to assign final grades on the basis of this exam only, above 81 would be an A (19), 65–80 a B (48), 40–64 a C (38), 21–39 a D (13), below 21 an E (3). Every student should have been able to get full credit on the second and third problems, plus a few points on each of the other problems; thus no score should have been below 50.

Problem Solutions

- (a) The proof of New Lemma 19.1 is *exactly* the same as the first two sentences of the proof of the original version.
- (b) The proof of New Lemma 19.4, which includes the observation that $s_0 = 1$ and $s_1 = 2$, begins the same as the proof of the original version, up to the phrase “To bound s_k , we count...”. Then we continue: To bound s_k , we count one for z itself and add the degrees of its children:

$$\text{SIZE}(x) \geq s_k \geq 1 + \sum_{i=1}^k s_{\text{DEGREE}(y_i)} \geq 1 + \sum_{i=1}^k s_{i-1} = 1 + \sum_{i=0}^{k-1} s_i.$$

It now follows by induction that $s_k \geq 2^k$.

- (c) Because $\text{SIZE}(x) \geq 2^{\text{DEGREE}(x)}$, taking logarithms base 2 we get $\lg \text{SIZE}(x) \geq \text{DEGREE}(x)$, so an n -element (modified) Fibonacci heap has $D(n) \leq \lfloor \log n \rfloor$.
2. This is an extraordinarily easy question! If there is any back edge, there is a cycle and hence the graph cannot be topologically sorted. Tree edges give the topological order; independent parts of the graph are not connected by any edges. Cross edges and forward edges can be there, but they play no role—that is, they don’t affect the topological sort.
3. The verification consists of three steps requiring a total of $O(|V| + |E|)$ time.

- (i) Verify that $s.d = 0$ and $s.\pi = \text{NIL}$. This takes $O(1)$ time.
- (ii) For all $v \in V$, $v \neq s$, check that $v.d = v.\pi.d + w(v.\pi, v)$ and that $v.d = \infty$ if and only if $v.\pi = \text{NIL}$. This takes $O(|V|)$ time.
- (iii) Finally, relax along each edge; none of these relaxations should change a distance value. This takes $O(|E|)$ time.

Failure of any step means the results are wrong; the vertices or edges where steps fail indicate places where the algorithm has produced nonsense.

4. Given an algorithm for HAM-CYCLE, we can use it to solve a HAM-PATH problem on graph G by adding a single vertex s which has an edge to every vertex of G ; call this augmented graph G' . Now if G' has a Hamiltonian cycle if and only if G has a Hamiltonian path, so solving HAM-CYCLE on G' is an algorithm for HAM-PATH on G .

On the other hand, given an algorithm for HAM-PATH, we can use it to solve a HAM-CYCLE problem on graph G as follows: If G has a Hamiltonian cycle containing edge e , the G' obtained from G by deleting e has a Hamiltonian path (that starts one of the endpoints of e and ends at the other). We try this test for every edge of G ; if none of them yields a Hamiltonian path, G does not contain a Hamiltonian cycle. This requires at most $|V|(|V| - 1)/2$ applications of the HAM-PATH algorithm—a polynomial number.

5. The last positions (those columns corresponding to the clauses) *must* have a 4 in the target number because we need at least one true variable in each (a 1 in the literal's row). So if we allowed the last positions in the target to be 1, 2, or 3 we could get those values using only the slack rows with all the literals being 0 (false). The *only* way to get a 4 is to have at least one literal be true.