MC57 Final Exam Solutions

Please note that these are sample solutions only; other solutions may be equally correct.

1. [**12 Points**] Consider the problem of expressing a non-negative integer as a sum of cubes of positive integers. For example, $1 = 1^3$, which is a sum of a single cube, while $2 = 1^3 + 1^3$ and $9 = 2^3 + 1^3$, so both 2 and 9 can be written as a sum of two cubes. We say that 0 can be written as a sum of zero cubes, since the empty sum is equal to 0. Let $W(n)$ be the smallest number of cubes of positive integers it takes to add up to n. So, $W(8) = 1$ because there is a single cube that is equal to 8, but $W(9) = 2$ because there is no positive integer which when cubed yields 9, but there is a sum of two cubes that equals 9, as we saw above. Show how to use dynamic programming to compute $W(n)$, and give an asymptotic bound on your algorithm's running time.

Answer:

```
W(n)
   T <- new array[0..n]
   T[0] <- 0
   for i <- 1 to n
      minSoFar <- i
      j <- 1
      while j^3 <= i
         minSoFar <- min(minSoFar, T[i - j^3] + 1)
         j <- j + 1
      T[i] <- minSoFar
   return T[n]
```

The outer loop is done n times, and the inner loop is done roughly $i^{1/3}$ times, so the total running time is $\Theta(\sum_{i=1}^{n} i^{1/3}) = \Theta(n^{4/3})$.

2. [**12 Points**] Suppose a minimum spanning tree has been found for the weighted undirected graph $G = (V, E)$, and that the set of edges constituting the tree is called T, where $T \subseteq E$. Now someone adds an additional edge to the graph; that is, they produce the graph $G' = (V, E \cup \{(u, v)\})$, where $u \in V$, $v \in V$, and $(u, v) \notin E$. Naturally, they also specify the new edge's weight, $w(u, v)$.

   The problem at hand is, how can the minimum spanning tree be updated to correspond to the new graph? That is, how can a minimum spanning tree for G' be found in a way that takes advantage of the knowledge that T is a minimum spanning tree of G? I propose the following outline of an algorithm for solving this problem:

   - From the edges making up the cycle in $(V, T \cup \{(u, v)\})$, choose one of maximal weight. Call this edge e.
   - The set $(T \cup \{(u, v)\}) - \{e\}$ is a minimum spanning tree of G'.

   There are several questions left open by this proposal; try to answer each of them. Note that these questions are largely independent, you need not work on them sequentially.

   (a) The first step speaks of "the cycle in in $(V, T \cup \{(u, v)\})$." This only makes sense if there is in fact a unique cycle in in that graph. Show that there is.

   (b) For the algorithm to be plausible, it is necessary that $(T \cup \{(u, v)\}) - \{e\}$ be a spanning tree of G'. Show that it is.

(c) For the algorithm to be correct, it is necessary that no other spanning tree of G' have a smaller total weight. Show that if there were a spanning tree of G' with smaller total weight, you could construct a spanning tree of G with total weight less than that of T, which would contradict the assumption that T is a minimum spanning tree of G.

(d) Finally, it is necessary to consider the efficiency of the proposed algorithm. Professor Scrooge claims that the algorithm can be implemented in such a way as to take time that is $O(|V|)$, while Professor Warbucks claims that any implementation of this algorithm would necessarily take time that is $\Omega(|E|)$ in the worst case. Who is right? Why?

Answer:

(a) Because T is a spanning tree, there is a unique path in T from u to v. This path plus the added edge (u, v) constitutes a cycle. There can't be any other cycle in $T \cup \{(u, v)\}$, because any cycle that didn't include (u, v) would also be a cycle in in T, which is impossible, and any that does include (u, v) implies the existence of another path in T between u and v, which is likewise impossible.

(b) If $e = (u, v)$, then $(T \cup \{(u, v)\}) - \{e\} = T$, so it is a spanning tree by assumption. If, on the other hand, $e \neq (u, v)$, then $(T \cup \{(u, v)\}) - \{e\} = (T - \{e\}) \cup \{(u, v)\}$. Removing e from T divides it into two connected components (because T is a spanning tree), and u and v are not in the same component (because e was on the path between u and v). Adding (u, v) reconnects these two components; therefore, it forms a new spanning tree.

(c) Suppose there were a spanning tree of G' (let's call it T') that is lighter than $(T \cup \{(u, v)\}) - \{e\}$. If $(u, v) \notin T'$, then $T' \subseteq E$, and so T' is a spanning tree of G. However, it is lighter than $(T \cup \{(u, v)\}) - \{e\}$, which is in turn at least as light as T, since $w(e) \geq w(u, v)$, so it would contradict T's minimality. Therefore, we can assume that $(u, v) \in T'$.

Since T' is a spanning tree, it is acyclic, and so it cannot contain all the edges of the cycle that e was chosen from. Let e' be one of the edges of that cycle not contained in T'. Since e was chosen to be maximal, $w(e') \leq w(e)$. Consider $(T' \cup \{e'\}) - \{(u, v)\}$. Since $e' \notin T'$, reasoning similar to that used in the previous part of the problem allows us to conclude this is a spanning tree of G. Its total weight, $w((T' \cup \{e'\}) - \{(u, v)\})$, is equal to $w(T') + w(e') - w(u, v)$. By assumption, $w(T') < w((T \cup \{(u, v)\}) - \{e\})$, so we have $w((T' \cup \{e'\}) - \{(u, v)\}) < w((T \cup \{(u, v)\}) - \{e\}) + w(e') - w(u, v) \leq w((T \cup \{(u, v)\}) - \{e\}) + w(e) - w(u, v) = w(T)$. Thus we appear to have found a spanning tree of G that is lighter than T, a contradiction. Thus T' must not exist.

(d) To locate the edge e, only edges in $T \cup \{(u, v)\}$ need to be considered, and there are only $|V|$ such edges. All the rest of the edges in E can be ignored totally. Thus it seems that Professor Scrooge, rather than Professor Warbucks, must be right.

3. [**12 Points**] Write a synopsis of one of the student presentations, other than your own. Try to cover the following points in your synopsis:

- What problem was being solved?
- What was the name of the algorithm presented for solving that problem?
- What were the key ideas of the algorithm itself?
- If a proof of correctness was sketched by the presenters, what were the key ideas underlying the proof?

- If an analysis of running time or other resource consumption was given, what were the key ideas underlying the analysis? What was the result of the analysis?

Answer:
One presentation showed how to multiply two $n \times n$ matrices using Strassen's algorithm, where n was a power of two. [Other sizes of matrices were handled by embedding them in such power-of-two square matrices.]
The key idea was to compute the four $n/2 \times n/2$ submatrices of the result in a way that used only seven multiplications of $n/2 \times n/2$ matrices rather than eight.
No explicit proof of correctness was given, but it was indicated how the correctness of one of the submatrices of the result could be confirmed by algebraic means; presumably the same would be true for the others.
The analysis of running time was done using the master theorem for divide and conquer recurrences. Since there were seven half-size subproblems, and the various sums and differences and submatrix extraction and embedding operations can be done in $\Theta(n^2)$ time, the recurrence was $T(n) = 7T(n/2) = \Theta(n^2)$. The solution was therefore $T(n) = \Theta(n^{\log_2 7})$.