Algorithms: MTH 524
Department of Mathematics and Statistics
Indian Institute of Technology - Kanpur February 6, 2015

Problem Set 2

Exercises are for extra practice and should not be turned in.

Exercise 1 *7.2-1 from CLRS :* Use the substitution method to prove that the recurrence $T(n) = T(n-1) + \Theta(n)$ has the solution $T(n) = n^2$.

Exercise 2 *15.2-1 from CLRS :* Find an optimal parenthesization of a matrix-chain product whose sequence of dimensions is $(5, 10, 3, 12, 5, 50, 6)$.

Exercise 3 *15.2-5 from CLRS :* Let $R[i, j]$ be the number of times that table entry $m[i, j]$ is referenced while computing other table entries in a call of MATRIX-CHAIN-ORDER. Show that the total number of references for the entire table is

$$\sum_{i=1}^{n} \sum_{j=1}^{n} R(i, j) = \frac{n^3 - n}{3}.$$

Note that this implies that the running time of MATRIX-CHAIN-ORDER is in fact $\Omega(n^3)$.

Exercise 4 *15.3-2 from CLRS :* Explain why memoization fails to speed up a good divide-and-conquer algorithm such as MERGE-SORT.

Exercise 5 *15.6 from CLRS :* Professor Stewart is consulting for the president of a corporation that is planning a company party. The company has a hierarchical structure; that is, the supervisor relation forms a tree rooted at the president. The personnel office has ranked each employee with a conviviality rating, which is a real number. In order to make the party fun for all attendees, the president does not want both an employee and his or her immediate supervisor to attend.

Professor Stewart is given the tree that describes the structure of the corporation, using the left-child, right-sibling representation described in Section 10.4. Each node of the tree holds, in addition to the pointers, the name of an employee and that employees conviviality ranking. Describe an algorithm to make up a guest list that maximizes the sum of the conviviality ratings of the guests. Analyze the running time of your algorithm.

Exercise 6 *16.1-4 from CLRS :* Suppose that we have a set of activities to schedule among a large number of lecture halls, where any activity can take place in any lecture hall. We wish to schedule all the activities using as few lecture halls as possible. Give an efficient greedy algorithm to determine which activity should use which lecture hall.

Exercise 7 *16.2-5 from CLRS :* Describe an efficient algorithm that, given a set $\{x_1, x_2, \ldots, x_n\}$ of points on the real line, determines the smallest set of unit-length closed intervals that contains all of the given points. Argue that your algorithm is correct.

Exercise 8 *22.2-4 from CLRS :* What is the running time of BFS if we represent its input graph by an adjacency matrix and modify the algorithm to handle this form of input?

Exercise 9 *24.1-3 from CLRS :* Given a weighted, directed graph $G = (V, E)$ with no negative-weight cycles, let m be the maximum over all vertices $v \in V$ of the minimum number of edges in a shortest path from the source s to v. (Here, the shortest path is by weight, not the number of edges.) Suggest a simple change to the Bellman-Ford algorithm that allows it to terminate in $m+1$ passes, even if m is not known in advance.

Exercise 10 *24.1-4 from CLRS :* Modify the Bellman-Ford algorithm so that it sets $d[v]$ to $-\infty$ for all vertices v for which there is a negative-weight cycle on some path from the source to v.

Exercise 11 *24.3-3 from CLRS :* Suppose we change line 6 of Dijkstra's algorithm (see lecture slides for details) to the following.

```
6    while $|Q| > 1$
```

This change causes the while loop to execute $|V| - 1$ times instead of $|V|$ times. Is this proposed algorithm correct?

Exercise 12 *24.3-6 from CLRS :* We are given a directed graph $G = (V, E)$ on which each edge $(u, v) \in E$ has an associated value $r(u, v)$, which is a real number in the range $0 \leq r(u, v) \leq 1$ that represents the reliability of a communication channel from vertex u to vertex v. We interpret $r(u, v)$ as the probability that the channel from u to v will not fail, and we assume that these probabilities are independent. Give an efficient algorithm to find the most reliable path between two given vertices.

Exercise 13 *24.3-8 from CLRS :* Let $G = (V, E)$ be a weighted, directed graph with nonnegative weight function $w : E \to \{1, \ldots, W\}$ for some nonnegative integer W . Modify Dijkstra's algorithm to compute the shortest paths from a given source vertex s in $O(WV + E)$ time.

Exercise 14 *24.3-9 from CLRS :* Modify your algorithm from *Exercise 24.3-8* to run in $O((V + E) \lg W)$ time. (Hint: How many distinct shortest-path estimates can there be in $V \setminus S$ at any point in time?)

The following problem is due **Friday, February 27** at **11:59 PM**.
Steps you should follow :

1. You should do all your work for this problem set in a directory called GroupID-PS2 where ID stands for your group number. For example, if you are homework group 05,

then the directory name will be Group05-PS1; if you are homework group 13, then the directory name will be Group13-PS1.

2. Upon completion of the work, once you are ready for submission, go the directory that contains the directory GroupID-PS2 and compress it using following command at the xterm/terminal command prompt

   ```
   $ tar cvfz GroupID-PS2.tgz GroupID-PS2
   ```

   which will create a file called GroupID-PS2.tgz .

3. Attach this file in an email with subject GroupID-PS2 and send it to akasha@iitk.ac.in by 11.59 pm on February 6. Late submission will not get any credit.

Let $\mathbb{P}_m(\mathbb{R}^n)$ denote the space of all real polynomials in n variable of degree m or less, i.e.,

$$\mathbb{P}_m(\mathbb{R}^n) = \left\{ p \mid p(x) = \sum_{|\ell| \le m} c_\ell x^\ell, c_\ell \in \mathbb{R} \right\}.$$

Note that for $\ell = (\ell_1, \ell_2, \ldots, \ell_n)$ with $\ell_i \in \mathbb{N}$, we denote $|\ell| = \sum_{i=1}^n \ell_i$ and $x^\ell = x_1^{\ell_1} x_2^{\ell_2} \ldots x_n^{\ell_n}$. For $p \in \mathbb{P}_m(\mathbb{R}^n)$, let $\Delta p \in \mathbb{P}_{m-2}(\mathbb{R}^n)$ be defined as

$$(\Delta p)(x) = \sum_{i=1}^n \frac{\partial^2}{\partial x_i^2} p(x)$$

and $\Delta^k p$ for $k > 1$ is defined inductively as $(\Delta^k p)(x) = (\Delta(\Delta^{k-1} p))(x)$.
Write an algorithm **COMPUTE-Q(p)** to compute the polynomial

$$q(x) = |x|^2 \sum_{k=0}^{[m/2]} \frac{(-1)^k |x|^{2k} \, (\Delta^k p)(x)}{(2,2)_{k+1} (n + 2m - 2k, 2)_{k+1}} \;, \tag{1}$$

for a given $p \in \mathbb{P}_m(\mathbb{R}^3)$, p is homogeneous (recall that a polynomial of degree j in n variables is called *homogeneous* if it satisfies $p(cx) = c^j p(x)$ for any $c \in \mathbb{R}$, $x \in \mathbb{R}^n$) and of degree m, where $[m/2]$ denotes the integer part of $m/2$, $x \in \mathbb{R}^n$ and $(a,b)_k = a(a+b)\cdots(a+(k-1)b)$ with $(a,b)_0 = 1$. Also recall that $|x|^2 = \sum_{i=1}^n x_i^2$ for $x \in \mathbb{R}^n$.
Implement your algorithm. The function prototype for **COMPUTE-Q(p)** should read

```
double compute-q(int m, double* coeff-p, double* x);
```

where the pointer `x` points to and array of size 3 holding a point of evaluation $x \in \mathbb{R}^3$ where `x[0]` $= x_1$, `x[1]` $= x_2$ and `x[2]` $= x_3$. The pointer `coeff-p` points to an array of size $(m+1)^3$ that contains the coefficients of the input polynomial p — the coefficient of $x_1^i x_2^j x_3^k$ is stored in `coeff-p[k + (m+1) * (j + (m+1) * i)]`. For example, if $p = 32x_1^4 - 5.5x_1^2x_2^2 + x_1x_2^2x_3 + 5x_2x_3^3 - 3x_1x_2x_3^2$ $(m = 4)$, the `coeff-p` would be of size `125` and be initialized as follows :

```
coeff-p[100] =  32;
coeff-p[60] =  -5.5;
coeff-p[36] =  1;
coeff-p[10] =  5;
coeff-p[32] =  -3;
```

with other elements of the array assigned the `0` (zero) value.
Remark : This way of representing a homogeneous polynomial in 3-variables is clearly not the most ideal for possibly sparse data. One can think about what kind of data-structures would be most suitable for representing a homogeneous polynomial in multiple variables, *but is* **not** *part for this problem set.*